

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Un logiciel de gestion des comptes pour personnes handicapées mentales développement et évaluation

Déplechin, Marc; Strappazzon, Gianni

*Award date:*  
1989

*Awarding institution:*  
Universite de Namur

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Année académique 1988-1989

**Un logiciel de gestion des comptes  
pour personnes handicapées mentales :  
développement et évaluation**

**Marc                      Gianni  
Déplechin              Strappazzon**

**Promoteur : Madame Monique Noirhomme-Fraiture**

**Co-promoteur : Monsieur Michel Mercier, professeur au  
Département de Psychologie, F.N.D.P.**

**Mémoire présenté en vue de  
l'obtention du grade de Licencié  
et Maître en Informatique**

## **Abstract**

Ce travail présente un projet de logiciel de gestion de budget pour personnes handicapées mentales. Le développement du programme suit une approche itérative et pluridisciplinaire qui voit la collaboration des psychologues, des éducateurs et des informaticiens. La phase d'évaluation se base sur des questionnaires et des traces de manipulations qui proviennent des expérimentations menées dans diverses institutions.

This work presents a project of budget management software for mentally disabled people. The program's development follows an iterative and multi-disciplinary approach leading to the collaboration of psychologists, educators and computer specialists. The evaluation phase is based upon questionnaires and data logging from experimentations done in various establishments.

Nous tenons à remercier vivement madame Monique Noirhomme-Fraiture pour les conseils judicieux qu'elle n'a cessé de nous prodiguer et pour le soutien qu'elle n'a jamais manqué de nous apporter. Nous remercions également monsieur Michel Mercier pour le temps qu'il nous a consacré et la motivation qu'il a su nous communiquer.

Un grand merci à André Baechler pour les heures de travail et de détente que nous avons passées ensemble lors de nos stages à Monthey.

Nous tenons à remercier tout particulièrement Odette sans qui ce travail n'aurait pas été terminé. Merci à Jean-Pierre, Vincent, Jacques, Claudette et Etienne, qui ont eu la patience de nous supporter et la gentillesse de nous aider pendant ces douze derniers mois.

Merci également à tous les éducateurs et psychologues au sein des diverses institutions belges et suisses qui ont participé aux phases de spécification et d'évaluation du logiciel.

Nous remercions enfin toute personne ayant contribué de près ou de loin à l'élaboration de ce travail.

**Loi de Hofstadter : Il faut toujours plus de temps que prévu,  
même en tenant compte de la loi de Hofstadter.**

# Table des matières

<b>Introduction.....</b>	<b>1</b>
<b>Chapitre I. Les domaines "Informatique et handicap" et "Interface homme-machine".....</b>	<b>6</b>
Introduction.....	6
1.1. Informatique et handicap.....	6
1.1.1. Problème des "aides technologiques".....	6
1.1.2. Hétérogénéité de la population handicapée.....	7
1.1.3. Evaluation de l'emploi de l'informatique dans le domaine du handicap mental.....	8
1.1.3.1. Utilisation possible dans l'éducation de handicapés mentaux.....	8
1.1.3.2. Neutralité.....	9
1.1.3.3. Les aspects affectifs.....	10
1.1.3.4. Garder des traces.....	10
1.1.3.5. Le côté attrayant de l'ordinateur.....	11
1.1.3.6. Nécessité d'une approche spécifique.....	12
1.1.4. Quelques exemples de réalisations en matière d'informatique et handicap.....	13
1.2. Généralités sur l'interface homme-machine.....	15
1.2.1. Quelques définitions.....	15
1.2.1.1. Une application interactive.....	15
1.2.1.2. Une interface homme-machine.....	16
1.2.2. Les techniques d'interaction.....	16
1.2.2.1. Relevé des différentes techniques.....	16
1.2.2.2. Techniques utilisables pour des utilisateurs handicapés mentaux.....	19
1.2.3. La problématique des utilisateurs.....	20
1.2.3.1. Plusieurs interfaces pour les mêmes fonctionnalités.....	20
1.2.3.2. Une approche itérative.....	21
1.2.3.3. Modèles de l'utilisateur et critiques.....	22
1.2.4. Principes et règles.....	23
1.2.4.1. La séparation modulaire entre l'interface et les fonctionnalités.....	23

1.2.4.2. Principes de Hansen.....	23
1.2.4.3. Les 8 règles de Ben Schneiderman.....	24
1.2.5. Les outils disponibles.....	25
1.2.5.1. Les boîtes à outils.....	26
1.2.5.2. Les systèmes génériques.....	26

## **Chapitre II. Le contexte dans lequel le projet est né.... 28**

2.1. Les acteurs impliqués.....	28
2.1.1. Le PSINHA.....	28
2.1.2. Les institutions demandeuses et participantes belges.....	28
2.1.3. La FST.....	29
2.1.4. Les institutions participantes suisses.....	30
2.2. Le programme de gestion de budget réalisé sur PC.....	30
2.2.1. Sur le plan fonctionnel.....	31
2.2.2. Sur le plan de l'interface.....	32
2.3. Le Commodore Amiga.....	35
2.3.1. La gamme Amiga.....	35
2.3.2. Les capacités graphiques de l'Amiga.....	36
2.3.2.1. Caractéristiques générales de l'Amiga.....	36
2.3.2.2. Schéma fonctionnel de l'Amiga.....	38
2.3.3. Les configurations utilisées dans le projet.....	40

## **Chapitre III. La démarche de spécification..... 42**

Introduction.....	42
3.1. Remarques méthodologiques.....	42
3.1.1. Les 4 idées de base.....	42
3.1.1.1. Analyser les fonctionnalités.....	42
3.1.1.2. Connaître les utilisateurs.....	43
3.1.1.3. Etudier les interfaces.....	43
3.1.1.4. Suivre une approche itérative.....	43
3.1.2. Une méthodologie de spécification d'une interface homme-machine.....	44
3.1.2.1. Distinction entre tâche et présentation.....	44
3.1.2.2. Spécification de l'interface scindée en deux parties.....	45

3.1.2.3. Applicabilité de la méthodologie dans notre projet.....	45
3.1.3. Les spécifications des fonctionnalités et de l'interface en parallèle.....	46
3.2. La participation des utilisateurs.....	48
3.2.1. Un problème de communication.....	48
3.2.2. Le rôle des psychologues.....	49
3.2.2.1. Psychologues actifs, concepteurs passifs.....	49
3.2.2.2. Remarques sur notre passivité.....	49
3.2.2.3. La collaboration souhaitée.....	50
3.2.3. Le rôle des utilisateurs.....	50
3.2.3.1. Prise de contact.....	51
3.2.3.2. La collaboration possible.....	51
3.3. La démarche suivie.....	52
3.3.1. La démarche.....	52
3.3.2. Commentaires sur la démarche.....	53
3.4. Comment s'est passée concrètement cette phase.....	54
3.4.1. En Belgique.....	54
3.4.2. En Suisse.....	55
3.4.2.1. Passifs.....	55
3.4.2.2. Actifs.....	58
3.4.2.3. Tests.....	61

## **Chapitre IV. Les spécifications du programme.....65**

Introduction.....	65
4.1. Les buts et les limites du programme.....	65
4.1.1. D'un point de vue pédagogique.....	65
4.1.1.1. Le projet répond à quels besoins ?.....	65
4.1.1.2. Quels sont les objectifs ?.....	66
4.1.1.3. A quelle population le projet est-il destiné ?.....	67
4.1.1.4. Quels sont les prérequis ?.....	67
4.1.1.5. Dans quel cadre sera-t-il utilisé ?.....	68
4.1.2. Limites de validité de tout programme d'aide à la gestion d'argent.....	69
4.2. Les spécifications du programme.....	70
4.2.1. Les spécifications des fonctionnalités.....	70
4.2.1.1. La tenue des comptes.....	72
4.2.1.2. Le catalogue.....	75



4.2.1.3. Le magasin.....	75
4.2.1.4. L'épargne et le budget.....	76
4.2.2. La spécification des interfaces abstraites (la tâche).....	76
4.2.3. La spécification des interfaces concrètes (la présentation).....	80
4.3. Les spécifications du programme de paramétrisation.....	92
4.3.1. Remarques préliminaires.....	92
4.3.2. Deux aspects contradictoires.....	93
4.3.3. Les fonctionnalités du programme.....	93
4.4. En quoi les règles et principes généraux d'interface respectés ?.....	96
4.4.1. Respect des 8 règles d'or de Ben Schneiderman.....	96
4.4.2. Respect des principes de Hansen.....	99

## **Chapitre V. L'architecture et le principe de séparation modulaire..... 101**

Introduction.....	101
5.1. Respecter le principe de séparation modulaire.....	101
5.2. L'architecture logique.....	102
5.2.1. Hiérarchisation et découpe en modules.....	102
5.2.2. Spécifications externes des modules.....	105
5.3. A propos de l'architecture physique.....	107
5.3.1. Modula-2.....	108
5.3.1.1. Particularités du Modula-2.....	108
5.3.2. Choix d'une architecture physique.....	109
5.3.3. Remarques sur l'environnement de programmation....	112

## **Chapitre VI. La phase d'évaluation.....113**

Introduction.....	113
6.1. L'état de l'art en matière d'évaluation de logiciels.....	113
6.2. Les deux types d'évaluation préconisés.....	116
6.2.1. Le point de vue interface.....	117
6.2.2. Le point de vue pédagogique.....	117
6.3. Les moyens à notre disposition.....	118
6.3.1. Les traces de manipulations.....	119
6.3.1.1. Pourquoi des traces de manipulations ?.....	119
6.3.1.2. Que contiennent les traces ?.....	119

6.3.1.3. Quelles informations peuvent-elles fournir ?.....	122
6.3.1.4. A qui les traces peuvent-elles être utiles ?.....	123
6.3.1.5. Graphe des états du programme de tenue des comptes.....	123
6.3.1.6. La matrice des transitions.....	126
6.3.1.7. Les transitions critiques.....	127
6.3.2. Les questionnaires.....	129
6.3.2.1. L'évaluation pédagogique.....	129
6.3.2.2. L'évaluation de l'interface.....	131
6.4. Les expérimentations.....	138
6.4.1. L'exploitation des traces.....	138
6.4.2. L'exploitation des questionnaires.....	143
6.4.2.1. L'analyse statistique.....	143
6.4.2.2. L'analyse en composantes principales.....	146

## **Chapitre VII. Conclusions et perspectives.....151**

7.1. Phase de spécification du projet.....	151
7.2. Phase d'évaluation.....	152
7.2.1. Les traces de manipulations.....	152
7.2.2. Les questionnaires.....	153
7.3. Perspectives au niveau du programme.....	154
7.4. Perspectives au niveau de l'évaluation.....	158
7.4.1. Plan d'expérimentation.....	158
7.4.2. L'exploitation des traces et des questionnaires.....	161

## Introduction.

Si l'informatique est restée longtemps réservée aux informaticiens, dans les dix dernières années, elle n'a cessé de s'ouvrir vers d'autres catégories de personnes. Elle a pénétré la plupart des milieux professionnels et a conquis même nos foyers.

Cette ouverture a considérablement élargi le champ des utilisateurs, et ceci a entraîné de nouveaux domaines à explorer en informatique. En effet, s'il est toujours crucial de concevoir des programmes qui répondent aux spécifications et qui réalisent correctement leurs fonctionnalités, il faut également qu'ils soient conviviaux et adaptés aux utilisateurs. C'est l'ordinateur qui doit s'adapter au mieux à l'homme et non l'inverse.

Bien sûr il reste toujours une certaine adaptation à la machine, mais l'idée est de faire faire le plus gros effort à cette dernière. Cette adaptation est prise en charge par l'interface homme-machine.

Nous avons dit que le champ des utilisateurs s'est considérablement élargi dans les dernières années. Ainsi dans la vie professionnelle, toute personne, quel que soit son niveau de qualification, peut être appelée à utiliser l'informatique à son poste de travail. Et dans la vie privée, on trouve des utilisateurs de "7 à 77 ans".

En quelques années, nous nous sommes habitués à cette nouvelle situation, par contre il n'est pas fréquent de destiner des produits informatiques à des utilisateurs handicapés. Pour ces utilisateurs, le problème de l'interface homme-machine est vital.

Dans le cas d'un handicap physique, les adaptations seront plus de type matériel, tandis que dans le cas d'un handicap mental les solutions seront de type logiciel. En tout cas, quel que soit le handicap de l'utilisateur, un gros effort devra être fourni au niveau de l'interface homme-machine, elle est primordiale et devra être particulièrement soignée.

Le problème de l'interface homme-machine nécessite déjà une

approche pluridisciplinaire, dès lors avec des utilisateurs handicapés cette approche est une condition sine qua non à tout travail sérieux. En effet, les connaissances mises en oeuvre relèvent de l'art graphique et sonore, des techniques d'affichage de données, des techniques d'interaction, de la compréhension des capacités motrices, perceptuelles et cognitives de l'être humain qui est face à l'ordinateur. Donc il est nécessaire de faire collaborer des personnes travaillant en informatique, en ergonomie et en psychologie. Et malgré tout rien ne garantit un résultat correct au premier essai. Dès lors, le développement d'un projet sera inévitablement un processus itératif où l'on évalue chaque sous-produit réalisé afin d'en découvrir les lacunes et d'en spécifier des modifications éventuelles.

Le problème de l'interface homme-machine, qui est loin d'être simple pour les applications interactives classiques, est donc encore plus complexe lorsque les utilisateurs sont handicapés. Toutefois, la phase de spécification de l'interface n'est pas la seule étape du développement qui nécessite une approche spécifique. La phase de spécification des fonctionnalités de l'application est déjà particulière. En effet, l'utilisateur handicapé n'exprime pas directement ses besoins. Dans la vie professionnelle, par exemple, les futurs utilisateurs d'une application peuvent participer à la définition des besoins et des spécifications de l'application. Ils peuvent être à la base du projet et avoir demandé explicitement la réalisation de celui-ci. Par contre, dans le cas d'applications pour personnes avec un handicap mental, cette situation ne se produit pas. Le projet ne part pas d'une demande directe des futurs utilisateurs. La demande émanera probablement des psychologues, des éducateurs, des animateurs,...

Il apparaît donc dans ce cas un type d'acteurs intermédiaire entre les informaticiens et les utilisateurs finals. Ces acteurs joueront un rôle important dans la phase de spécification des fonctionnalités de l'application interactive à réaliser.

Néanmoins, même si les utilisateurs finals ne formulent pas de demande initiale, ils doivent d'une façon ou d'une autre participer au travail de spécification. Cette participation dès le début du projet est, selon nous, absolument nécessaire. Nous verrons comment tout cela s'est passé dans le projet particulier qui nous a occupé.

Enfin, outre la nécessité d'une approche spécifique pour la phase de spécification des fonctionnalités et pour la phase de spécification de l'interface homme-machine, il faut également apporter un soin particulier à la phase d'évaluation du logiciel. En effet, comme c'est le cas pour la phase de spécification, se pose le problème de la communication avec l'utilisateur final. Il ne peut lui être demandé de faire une évaluation explicite d'un logiciel, de faire des critiques écrites, etc. Ce sera avant tout par l'intermédiaire de l'accompagnant que seront exprimées les critiques, les propositions qui seront à la base des itérations dont nous avons parlé plus haut. Nous verrons comment prendre en charge l'évaluation en collaboration avec les accompagnants, mais aussi comment il est tout de même possible de prendre en considération les "critiques implicites" qui émanent des utilisateurs. Il s'agira en fait d'analyser des traces d'utilisation qui sont enregistrées après chaque session de travail. Nous verrons également les deux types d'évaluation qui peuvent être faites d'un programme pour utilisateurs handicapés.

L'objectif de ce travail est de présenter la problématique du développement de projets informatiques destinés à des utilisateurs avec un handicap mental. Nous le ferons essentiellement en nous basant sur le projet que nous avons mené, en exposant de façon pragmatique la démarche que nous avons suivie lors des différentes phases. Nous nous efforcerons de mettre en lumière les forces et les faiblesses évidentes. Il ne s'agit en aucun cas d'une méthodologie à suivre pour travailler dans le domaine de l'informatique et du handicap mental, mais plus d'une expérience qui pourrait susciter d'autres travaux plus théoriques visant à proposer une méthodologie pour mener à bien les différentes étapes du cycle de vie d'un projet informatique dans le domaine des handicapés mentaux.

Le chapitre 1 présentera les travaux existants dans le domaine de l'informatique et du handicap en général (mental et physique), ainsi qu'un bref aperçu de l'état de l'art et des recherches en matière d'interface homme-machine (puisque, si une approche spécifique est nécessaire pour des utilisateurs handicapés mentaux, il n'en reste pas moins vrai que bon nombre de principes généraux de l'interface homme-machine restent

toujours d'application).

Le **chapitre 2** exposera le contexte dans lequel ce projet est né, en passant en revue les différents acteurs impliqués, les outils utilisés et le travail qui a servi de base au nôtre. Il s'agit d'un programme d'aide à la gestion de budget pour personnes handicapées mentales réalisé sur Schneider PC 1512 Amstrad.

Le **chapitre 3** présentera la démarche que nous avons suivie lors de la phase de spécification. Nous verrons que la spécification des fonctionnalités et celle de l'interface homme-machine ne sont pas indépendantes et nous verrons comment elles sont menées parallèlement dans un processus essentiellement itératif. Nous insisterons sur le fait que, dès cette phase, la participation des psychologues et des utilisateurs est nécessaire.

Dans le **chapitre 4**, nous exposerons en détail les spécifications que nous avons élaborées et nous définirons la partie de celles-ci qu'il était raisonnable de développer dans le temps qui nous était imparti, à savoir un programme d'aide à la tenue des comptes.

Le **chapitre 5** exposera la phase de conception du programme et son architecture qui s'appuie sur le principe fondamental de séparation modulaire entre les fonctionnalités et l'interface proprement dite.

Dans le **chapitre 6**, nous exposerons les deux types d'évaluation que nous préconisons pour un programme interactif destiné à des handicapés mentaux. Ces deux évaluations peuvent être réalisées de deux façons complémentaires : il s'agit, d'une part, de l'exploitation des traces gardées lors de toute session de travail et, d'autre part, de la construction et l'exploitation de grilles d'évaluation complétées par les éducateurs, accompagnants,...

Ce chapitre présentera également la phase d'expérimentation qui s'est déroulée dans deux institutions pour handicapés mentaux. Nous verrons brièvement les éléments dont il faut tenir compte pour l'exploitation des résultats.

Le chapitre 7 passera en revue les points forts et les faiblesses de notre démarche au cours de toutes les étapes du cycle de vie du projet, et proposera un certain nombre de voies à explorer dans des travaux ultérieurs.

# Chapitre 1. Les domaines "Informatique et handicap" et "Interface homme-machine"

## Introduction

Le but de ce chapitre est de voir quelles sont les spécificités du milieu des handicapés mentaux qui font du développement d'un projet informatique une activité particulière. Nous verrons également quelles sont les réflexions, les attentes des psychologues vis-à-vis de l'informatique et nous présenterons ce qui se fait en matière d'informatique et handicap.

Ce chapitre présentera le problème de l'interface homme-machine en toute généralité, ainsi que les différentes techniques d'interaction.

## 1.1. Informatique et Handicap

### 1.1.1. Problème des "aides technologiques"

L'informatique et les autres nouvelles technologies peuvent être mises en oeuvre pour chercher à aider les personnes handicapées. Elles peuvent servir à compenser un handicap et permettre à ces personnes de réaliser des activités jusque là impossibles à cause de leur handicap moteur, sensoriel ou intellectuel. Mais il faut être conscient de la problématique des "aides technologiques" (elles concernent tous les types de handicap). En effet, des savants, des chercheurs, des médecins, des psychologues, des thérapeutes essaient de mettre au point des techniques, des programmes, des "prothèses" dans le but d'aider, mais ce n'est pas parce qu'un appareil ou un logiciel est conçu pour aider qu'il apportera obligatoirement une véritable aide. Il est parfois difficile pour un chercheur qui a consacré beaucoup de travail et de temps à mettre au point une prothèse qui lui semble idéale dans son système de référence, de comprendre pourquoi cette prothèse est refusée par le patient [COURBIN]. Il en va de même pour les "aides technologiques" de type logiciel. Cette situation est une réalité dans le domaine de l'informatique et du handicap. Nous avons souvent été confrontés à cette situation dans le projet que nous avons mené. Nous avons pu remarquer que souvent, les personnes



handicapées mentales ne suivaient pas le même schéma logique que les personnes non-handicapées. Il faut donc être toujours très prudent et ne jamais oublier que ce qui semble être une "prothèse idéale" pour aider les personnes handicapées ne leur sera peut-être d'aucune utilité car elle ne respectera pas LEUR logique.

### **1.1.2. Hétérogénéité de la population handicapée**

Du point précédent, il découle qu'une attention toute particulière doit être accordée aux futurs utilisateurs. Il est inutile de réaliser un programme, un appareil techniquement parfait si celui-ci est rejeté par l'utilisateur auquel il était destiné. Par conséquent, tous les projets liant informatique et handicap en général, et en particulier ceux qui ont pour but la création de logiciels destinés à des handicapés mentaux doivent être menés en gardant bien à l'esprit que, pendant tout le processus d'élaboration, la technique ne doit pas prendre le pas sur la personne.

Donc, dans le cycle de vie d'un projet informatique, les utilisateurs doivent être impliqués dès les premières phases, en l'occurrence celle de spécification des fonctionnalités et de l'interface. Bien entendu, cette participation n'est pas si simple et nécessite une approche particulière vu les capacités des utilisateurs. L'éducateur ou le psychologue devient un acteur supplémentaire indispensable dans les différentes phases. Nous verrons cela plus en détail dans le chapitre suivant qui traite de la phase de spécification.

Prendre en compte les capacités, les besoins des utilisateurs n'est pas chose simple et ne garantit pas le succès, l'adéquation entre la machine (le programme) et l'homme. Le danger est de croire qu'il est possible de relever des caractéristiques communes à toutes les personnes handicapées. En effet, ce n'est pas parce que toutes ces personnes sont handicapées mentales qu'elles constituent un ensemble homogène. Bien que les psychologues et les médecins essaient de déterminer des classes de handicapés ayant plus ou moins les mêmes caractéristiques, il nous est apparu que chaque personne handicapée est un cas à part qu'il faut traiter à part. Si une population normale est constituée d'individus ayant des réactions différentes face à une même situation, cette divergence dans les réactions est bien plus présente dans une population handicapée mentale.

Nous verrons ainsi qu'il faut rester très très prudent lorsqu'on essaie de définir des classes d'utilisateurs pour mettre en oeuvre l'évaluation du programme (voir chapitres 5 et 6).

### 1.1.3. Evaluation de l'emploi de l'informatique dans le domaine du handicap mental

Le but de cette section est de relever les critiques faites par des psychologues sur l'emploi de l'outil informatique en éducation spécialisée. Donc, elles se basent sur des expériences réelles d'implantation de la micro-informatique dans diverses institutions accueillant des handicapés mentaux. Il est donc question ici d'évaluer dans une très faible mesure l'outil informatique dans ce contexte.

Bien entendu, l'informatique n'est ni bonne ni mauvaise en soi ; ce n'est qu'un outil qu'on utilise plus ou moins convenablement.

#### 1.1.3.1 Utilisation possible dans l'éducation de handicapés mentaux

Comme nous l'avons dit, l'informatique est un outil, mais il est clair qu'il n'est pas le seul, ni forcément le meilleur pour atteindre les buts recherchés. Il y a donc lieu de se questionner sur l'apport supplémentaire que pourrait constituer l'informatique par rapport aux autres solutions possibles. Néanmoins, cette réflexion ne doit pas permettre de rejeter l'informatique en bloc; cet examen de l'intérêt de l'utilisation de l'informatique doit être fait pour chaque application. Ainsi, l'informatique pourra être rejetée pour une certaine application et préférée pour une autre application.

Néanmoins, les perspectives de l'informatique en général dans le domaine des déficiences mentales semblent bonnes. Rappelons à ce propos les conclusions d'un groupe de travail "informatique et éducation" qui a été mis en place par l'UNAPEI (Union Nationale des Amis et Parents d'Enfants Inadaptés) qui rassemble plus de 1200 établissements de soins sur l'ensemble de la France : "Au stade atteint par la réflexion, à partir des expériences internes à l'Union et de celles d'autres organismes en France et à l'étranger, il semble possible de considérer que l'on peut donner une

réponse positive à la question posée par les administrateurs de l'Union : de préciser si l'idée d'utiliser l'informatique comme support à l'éducation d'enfants déficients mentaux est réaliste". On décèle dans cette conclusion une grande prudence, mais nous avons pour notre part rencontré en Belgique et en Suisse la confirmation qu'il est réaliste d'utiliser l'informatique dans l'éducation de personnes handicapées mentales. L'enthousiasme pour l'utilisation de l'informatique est grandissant et l'on peut remarquer que de plus en plus des demandes émanent des éducateurs et des psychologues pour des logiciels d'éducation spécialisée (comme nous le verrons dans le chapitre II, le projet sur lequel nous avons travaillé émane directement de deux institutions belges).

Nous allons maintenant essayer de voir plus spécifiquement quels sont les avantages de l'utilisation de l'ordinateur en éducation spécialisée.

#### 1.1.3.2. Neutralité

De par sa grande "neutralité", l'ordinateur est un moyen pour permettre à certaines personnes handicapées mentales de réaliser des tâches qu'ils n'étaient pas capables de faire avant l'introduction de la machine. Face à l'ordinateur, la personne handicapée ne se trouve plus dans une situation pédagogique de type hiérarchisé, c'est-à-dire une situation dans laquelle le handicapé sans connaissance se trouve face à un éducateur dont la connaissance lui paraît illimitée, éducateur qui va d'une façon ou d'une autre le pénaliser s'il commet des erreurs. Elle est au contraire engagée dans un dialogue (ce mot est peut-être mal choisi car le handicapé ne perçoit en aucune façon cette interaction comme un dialogue humain) avec une machine qui n'a aucun pouvoir de connaissance, de contrôle, de punition sur lui, ce qui peut la rassurer et lui permettre d'aller plus loin. La position d'échec qui est trop souvent celle d'une personne ayant un handicap peut ainsi être effacée, ou du moins amoindrie.

Ainsi, l'ordinateur est souvent perçu par les personnes handicapées mentales comme un "éducateur" (dans le sens "moyen d'éducation") moins sévère qu'un éducateur humain. La frustration qui peut résulter du dialogue avec l'éducateur qui "a toujours raison" disparaît lorsque le dialogue a lieu uniquement entre la personne handicapée et l'ordinateur.

Dans le cas qui nous occupe, et qui traite du problème de gestion de budget pour handicapés mentaux, cette neutralité est un atout certain. En effet, les handicapés mentaux, et principalement les enfants, sont pour la plupart égocentriques. Ainsi, un enfant refusera qu'une autre personne touche à - ou même voit - son argent. Dès lors, l'utilisation de l'ordinateur peut être bénéfique : l'utilisateur n'a pas peur que l'ordinateur lui prenne son argent, impression qu'il peut ressentir si son interlocuteur est humain. En ce sens, l'ordinateur est plus neutre que n'importe quelle personne, et c'est souvent un avantage important.

#### **1.1.3.3. Les aspects affectifs**

Les aspects affectifs sont très importants dans la vie quotidienne avec les handicapés mentaux. Nous avons vu que par son caractère neutre, l'informatique bien employée pouvait éviter des situations où l'utilisateur se sent brimé, inférieur à l'éducateur et son immense savoir. Il est aussi possible, grâce à des programmes adéquats, de renforcer chez l'utilisateur d'autres aspects affectifs tels que la confiance en soi (qui fait souvent défaut), la prise de décision et le fait de poser un choix et d'en accepter la responsabilité.

Il faut également souligner le plaisir ressenti par beaucoup de personnes handicapées qui sont mises en contact avec l'outil informatique. Ces personnes sont toujours très heureuses de "travailler avec l'ordinateur" et cette motivation est souvent supérieure à tout jeu que pourrait imaginer un éducateur.

Les adultes handicapés mentaux sont encore davantage valorisés que les enfants. Il arrive fréquemment que les éducateurs soient plus jeunes que les handicapés adultes dont ils s'occupent. Cette situation est ressentie comme gênante et oppressante et rend ces handicapés très conscients de leur dépendance. Travailler avec l'ordinateur atténue ce sentiment et leur donne l'impression d'être autonome et de "faire quelque chose tout seul".

#### **1.1.3.4. Garder des traces**

L'ordinateur a encore cet avantage de permettre très facilement un suivi des manipulations effectuées par une personne handicapée qui utilise

un programme. Ainsi, un programme peut avoir été conçu pour sauver, pendant toute la session, un certain nombre d'informations concernant les manipulations de l'utilisateur (il s'agit donc d'instructions ne gênant en aucun cas l'exécution des fonctionnalités propres du programme). Les informations contenues dans la trace peuvent bien entendu être très diverses, mais seules seront gardées celles qui ont un intérêt pour l'évaluation du programme comme par exemple des renseignements sur les opérations qui pourraient entraîner des erreurs de manipulation ou des renseignements sur la séquence des options successivement actionnées par l'utilisateur. Nous verrons dans le chapitre traitant du problème de l'évaluation du logiciel (chapitre 6) que ces traces constituent un des deux outils que nous avons employés pour cette phase. De ces traces, il y aura donc moyen de tirer des "critiques implicites" (le handicapé ne pouvant pas critiquer explicitement, il ne peut le faire qu'à travers l'utilisation du programme), c'est-à-dire qu'à travers les erreurs enregistrées lors de l'utilisation, il sera possible de mettre en lumière des incohérences du programme ou des inadéquations du dialogue avec l'utilisateur. Nous verrons plus en détail à quoi peuvent servir ces traces des manipulations et ce qu'elles peuvent contenir.

On voit donc l'intérêt de garder les traces des manipulations des utilisateurs lors de chaque session de travail : si elles contiennent des informations intéressantes, elles peuvent servir à évaluer et à améliorer un programme. Bien entendu, il n'est pas nécessaire de se servir d'un ordinateur pour garder des informations sur une session de travail mais cela permet d'éviter la présence permanente d'un accompagnant, cela va beaucoup plus vite et chaque trace aura la même forme et contiendra toujours les mêmes informations, ce qui pourrait ne pas être le cas avec une méthode manuelle.

#### 1.1.3.5. Le côté attrayant de l'ordinateur

Nous en avons déjà quelque peu parlé, il est indéniable que l'ordinateur est très souvent attrayant pour toute personne, handicapée ou non. On peut le voir en observant le succès que connaissent les jeux vidéo. Les formes, les couleurs, le son peuvent être employés pour attirer



l'attention et faire en sorte que les utilisateurs se sentent motivés, ressentent un plaisir certain à travailler avec l'ordinateur. L'aspect plaisir est un élément très important dont il faut tenir compte lorsqu'on envisage l'utilisation de l'informatique pour une tâche éducative quelconque avec les handicapés mentaux.

#### 1.1.3.6. Nécessité d'une approche spécifique

Dans les sections précédentes, nous avons donc relevé une vision globalement positive sur l'emploi des ordinateurs dans le monde de l'éducation pour handicapés mentaux. Nous avons ainsi pu observer une certaine neutralité de l'ordinateur qui permet notamment d'éviter certaines situations où le handicapé se trouve dans une position inférieure à celle de l'éducateur, une possibilité de renforcer peu à peu la confiance en soi et la prise de responsabilités, une facilité de garder des traces d'utilisation dans un but d'évaluation et d'amélioration du programme et finalement le côté attrayant de la machine.

Tout ceci semblerait dire que l'emploi d'ordinateurs et de programmes d'éducation est toujours une réussite et apporte un plus dans l'éducation et la vie de tous les jours des personnes handicapées mentales. C'est évidemment faux !

La neutralité dont nous avons parlé peut être mal perçue et l'utilisateur peut préférer le contact humain avec l'éducateur qu'il connaît mieux. Cela dépend certainement de la situation et des relations antérieures existant entre l'éducateur et les handicapés et la qualité du dialogue entre le programme et l'utilisateur.

Affectivement, au lieu de prendre confiance en soi, la personne peut au contraire être angoissée par l'utilisation de l'ordinateur et avoir peur de faire de fausses manipulations (cette situation est une réalité aussi bien dans la population non-handicapée que chez les personnes handicapées). La survenance ou la non-survenance de cette situation dépendra probablement de la façon dont la machine a été "présentée" aux utilisateurs et de la capacité du programme à reprendre les erreurs.

De plus, il est toujours possible qu'un programme apparemment attrayant puisque utilisant des couleurs, du graphique, du son ne soit pas

motivant pour certaines personnes handicapées. Lors de notre stage, nous avons rencontré certaines personnes handicapées qui n'étaient pas le moins du monde intéressées par ce qui se passait à l'écran.

Et bien entendu, outre ces remarques, il faut noter qu'un rejet sera total si le programme n'est pas adapté au public utilisateur.

Les programmes destinés à des personnes ayant un handicap mental posent donc des problèmes spécifiques qui demandent à être résolus par une approche différente de celle utilisée pour des programmes éducatifs standard. On peut, en effet, se baser sur son expérience personnelle pour construire des programmes standard ; mais pour des personnes, adultes ou enfants, ayant de grosses difficultés d'apprentissage, il est nécessaire de comprendre les capacités réelles des utilisateurs avant de chercher à concevoir des programmes. Le développement de programmes adaptés doit donc absolument partir de la connaissance des utilisateurs et s'appuyer sur l'expérience approfondie de l'éducation spécialisée qu'ont les éducateurs et les psychologues. De plus, parallèlement, le développement se basera également sur des principes généraux, comme ceux de l'interface homme-machine qui font l'objet du point 1.2.

#### 1.1.4. Quelques exemples de réalisations en matière d'informatique et handicap

Lorsqu'on regarde les travaux dans le domaine de l'informatique et du handicap, on remarque que la plupart sont axés sur le handicap physique. On peut citer la construction de James, une télécommande infrarouge permettant à une personne handicapée physique de diriger quantité d'instruments (lampes, chaîne hi-fi, portes, télévision,...). Cet appareil "apprend" et mémorise les signaux infrarouges émis par les différentes télécommandes, ce qui permet de n'avoir plus qu'un seul appareil de commande (appareil réalisé par la FST - Fondation Suisse pour les Téléthèses).

On le voit, le handicap physique fait l'objet de recherches importantes dans beaucoup de pays. Dans ce domaine, l'ordinateur servira souvent de prothèse qui va soit améliorer les capacités résiduelles d'une activité sensorielle ou motrice, soit transformer les informations inaccessibles en

raison du handicap par d'autres informations auxquelles il est possible d'avoir accès (il remplace le geste par la parole, la vue par le mot, la parole par l'écrit,...).

C'est dans cette catégorie de prothèses que l'on peut donc ranger les travaux existants en matière de handicap de la communication (Infirmités Motrices Cérébrales IMC avec ou sans troubles intellectuels). L'ordinateur peut alors permettre le dialogue à des personnes qui ont ces difficultés de communiquer ; il s'agit de communiquer "à travers" la machine. C'est donc principalement un outil prothèse. C'est le cas de HECTOR bien connu des personnes travaillant dans le milieu des IMC. HECTOR est un ordinateur disposant d'un clavier spécial dont les touches ont plus ou moins 3cm/3cm. Lorsque le handicapé appuie sur une touche, un mot est prononcé. Il peut ainsi, via cet ordinateur, atteindre un niveau de communication très satisfaisant. Les touches sont programmables, c'est-à-dire que l'on peut changer les mots qui y sont associés et grâce à des touches de fonction, à chaque touche peuvent être associés plusieurs mots. Grâce aux diverses prothèses de communication, on peut remarquer que certaines personnes sont petit à petit parvenues à communiquer sans aide technique. L'ordinateur a donc amené dans certains cas à un déblocage de la communication (bien entendu, il faut noter que les personnes qui font de tels progrès ont un niveau intellectuel relativement élevé, maîtrisent un vocabulaire assez riche, etc).

Dans le domaine du handicap mental par contre, il semble que les travaux soient beaucoup plus rares et plus dispersés. Lors d'un colloque formation "Informatique et Déficience Intellectuelle" qui a eu lieu à Lons-le-Saunier, une réflexion a été émise qui confirme ce point de vue : "ce colloque nous a paru riche en expériences diverses et nous a permis de découvrir les applications possibles de l'informatique. Cependant, dans le domaine de l'informatique lié à la déficience intellectuelle, tout reste à faire, et les différentes actions menées ici et là prennent des allures pionnières en la matière" [UNAPEI]. Il semble toutefois qu'un désir de collaboration soit en train de s'installer entre les équipes oeuvrant en la matière dans plusieurs pays francophones.

Dans le projet qui nous a occupés, comme nous le verrons plus loin, nous avons été en contact avec des équipes belges et suisses qui



collaborent également avec des équipes françaises, notamment celle de l'Université Technologique de Compiègne sous la direction de M. Le Cardinal. L'équipe belge était celle du PSINHA (PSychologie-INformatique-HAndicap) du département de psychologie des Facultés Universitaires Notre-Dame de la Paix de Namur et l'équipe de Suisse était celle de la FST (Fondation Suisse pour les Téléthèses) qui s'occupe davantage du handicap physique mais qui s'ouvre maintenant au handicap mental. Ces deux institutions seront présentées un peu plus longuement dans le chapitre 2.

## 1.2. Généralités sur l'interface homme-machine

Le but de cette partie est de voir brièvement les travaux, les recherches et les principes actuels en matière d'interface homme-machine. Ces considérations ne visent pas l'interaction avec l'homme handicapé mental, mais nous verrons qu'elles sont souvent tout à fait d'application (car les techniques d'interaction sont toujours les mêmes, les principes de conception restent valables, et les outils à la disposition des concepteurs sont inchangés). La différence se situe au niveau de l'utilisateur et de la façon dont on pourra communiquer avec lui durant les phases du cycle de vie d'un projet (cette problématique sera en toile de fond de tous les chapitres suivants).

Voyons donc les aspects généraux et plus ou moins invariants de l'interface homme-machine.

### 1.2.1. Quelques définitions

#### 1.2.1.1. Une application interactive

Une application interactive implique l'intervention de l'opérateur dans le traitement qu'il effectue. Par opposition, les systèmes "en temps différés" ne nécessitent pas cette intervention. On parlera d'une application interactive lorsqu'elle est caractérisée par la part importante de dialogue qu'elle nécessite avec l'utilisateur pour assurer l'exécution des fonctionnalités qu'elle offre [Chadelon & Warnant, 1987]. Cette définition met déjà en évidence deux éléments importants ; d'une part les

fonctionnalités qu'offre l'application et d'autre part l'interface ou le dialogue avec l'utilisateur.

Nous verrons comment il convient de séparer ces deux aspects à la conception de l'application (mais pas à la spécification).

#### 1.2.1.2. Une interface homme-machine

Au sens informatique du terme, une interface est l'ensemble des règles et conventions qui régit la communication entre deux systèmes organisés [Chernicoff, 1985].

Plus particulièrement, l'interface homme-machine règle le dialogue entre l'utilisateur et l'ordinateur [Petoud, 1986].

Dans le but de faciliter la tâche de l'utilisateur, il faut concevoir une interface qui se rapproche le plus possible de ses caractéristiques. Bien entendu, il n'est pas facile de mettre en lumière ces caractéristiques et dans le cas d'utilisateurs ayant un handicap mental, cette tâche est d'autant plus complexe et nécessite une approche spécifique. De toute façon, ceci fait présager que la conception d'une bonne interface homme-machine n'est pas chose aisée, et que les problèmes rencontrés relèvent de diverses disciplines, par conséquent il sera nécessaire de suivre une approche pluridisciplinaire.

Remarque : dorénavant nous utiliserons les termes "interface homme-machine", "interface utilisateur", "dialogue" comme des synonymes.

#### 1.2.2. Les techniques d'interaction

##### 1.2.2.1. Relevé des différentes techniques

La section précédente posait les définitions de base de la problématique, celle-ci énumère les techniques qui sont généralement utilisées dans les applications interactives classiques. Nous verrons parmi ces techniques celles qu'il est raisonnable de mettre en oeuvre pour une

application destinée à des personnes avec un handicap mental.

On peut classer les techniques d'interaction de différentes manières, voici un relevé de ces techniques (qui ne prétend pas être indiscutable ni exhaustif).

- \* un langage de commande : la méthode la plus traditionnelle pour donner des instructions à un système interactif est l'introduction de lignes de commandes. L'ensemble des commandes autorisées étant défini dans un langage de commande formel.
- \* un langage de programmation : l'ensemble des commandes que peut effectuer un utilisateur est parfois tellement vaste qu'il est nécessaire d'offrir un langage de programmation pour pouvoir communiquer avec l'application interactive. L'exemple type est celui des systèmes d'interrogation de base de données.
- \* le menu : par cette technique, l'utilisateur effectue des choix et déclenche des opérations sans qu'aucun langage de commande ou de programmation ne lui soit nécessaire. C'est donc une technique très intéressante pour des dialogues adaptés à des utilisateurs non spécialistes. D'autant plus qu'en combinant cette technique avec d'autres graphiques (voir plus loin), on peut donner une forme tout à fait simple à un menu.
- \* le remplissage de formulaire : par cette technique, l'utilisateur communique un certain nombre d'informations à l'application en remplissant un formulaire apparaissant à l'écran. La notion de formulaire pouvant être réduite à un seul "champ", une seule donnée que l'utilisateur doit introduire. Mais, bien entendu, un formulaire affiché à l'écran peut être beaucoup plus complexe et contenir un grand nombre de champs. Il peut correspondre plus ou moins précisément à un formulaire "papier" existant par ailleurs. Par exemple, pour un traitement de commande d'articles par correspondance, l'opérateur peut voir à l'écran un formulaire représentant exactement la structure du formulaire de commande qu'envoient les clients par la poste.

- \* les icônes : des commandes peuvent être représentées sous forme graphique. Dans ce cas l'utilisateur déclenchera une action en désignant le graphique correspondant (cette désignation sous-entend une technique de manipulation directe, voir plus loin). La technique graphique est aussi très utile en matière de feed-back : un feed-back est une information que le système fournit en réponse à une action de l'utilisateur. Donc suite à une action, un feed-back graphique peut être affiché sur une partie de l'écran (cette technique sera largement utilisée dans notre application).
- \* la manipulation directe : l'utilisateur manipule, à l'aide d'une souris ou d'un autre mécanisme de sélection, une représentation graphique. Il peut donc, comme nous l'avons vu, déclencher une action, il peut sélectionner une option d'un menu, etc.
- \* la fenêtre : une fonction spécifique peut être représentée à l'écran par une fenêtre (qui plus que probablement contiendra des éléments graphiques ou des formulaires). Un type particulier de fenêtre que nous avons utilisé est un "requester" (ou encore "boîte de dialogue modale"). Un requester apparaît à l'écran suite à une action et demande à l'utilisateur une réponse (souvent la confirmation d'une demande ou des informations nécessaires à la continuation du travail en cours) ; tant que l'utilisateur ne donne pas de réponse, il est impossible de continuer la session de travail. Il s'agit donc d'un type de fenêtre utilisé quand l'application a absolument besoin d'une information pour pouvoir avancer dans le travail.
- \* le son : le son constitue évidemment une technique d'interaction entre l'homme et la machine. Toutefois dans l'état actuel des choses, il est plus facilement utilisable dans le sens "machine -> homme", que l'inverse. Donc il pourra être utilisé pour poser des questions oralement à l'utilisateur, et pour donner un "feed-back acoustique" suite à une action. Il n'empêche qu'il existe des systèmes où une certaine reconnaissance de la parole est utilisée et où il est possible

à l'utilisateur de déclencher un certain nombre d'actions par la voix. Par exemple, dans le domaine des handicapés, nous avons vu un système où une personne alitée pouvait modifier l'orientation de son lit en disant un mot (l'ordinateur pouvant en reconnaître une dizaine). Il s'agit donc là de reconnaissance de mots isolés qui permet déjà une interaction dans le sens "homme -> machine" (et il y a évidemment moyen de faire mieux).

Nous avons donc relevé les diverses techniques. Bien entendu une application interactive utilisera plusieurs de ces techniques, en les combinant éventuellement selon le désir de l'utilisateur (nous en reparlerons dans les chapitres suivants).

#### **1.2.2.2. Techniques utilisables pour des utilisateurs handicapés mentaux**

Il n'est pas question de dire une fois pour toutes quelle technique peut ou ne peut pas être utilisée pour tel type de problème, car tout dépend de la gravité du handicap dont souffre l'utilisateur, et de l'implémentation particulière d'une technique d'interaction. Toutefois les techniques de langages de commandes ou de programmation nous semblent peu adaptées à des utilisateurs handicapés (il n'empêche que des langages de commandes peuvent être envisagés, par exemple on pourrait imaginer un langage où l'utilisateur n'introduit pas les commandes via le clavier, mais où il "compose" ses commandes à partir d'icônes qu'il combine).

La technique du menu, par contre, est tout à fait adaptée, surtout si elle combinée avec des icônes et de la manipulation directe. Ces trois techniques sont donc parfaitement utilisables.

Le remplissage de formulaires peut être envisagé, le tout étant de trouver la forme la plus adaptée possible aux caractéristiques de l'utilisateur. En tout cas, comme nous le verrons, il est tout à fait raisonnable de demander à des personnes handicapées mentales d'introduire une information via le clavier, et cela même pour des personnes qui ne savent pas lire (par exemple en recopiant un modèle, nous en reparlerons, notamment avec les tickets de caisses).

La technique d'interaction utilisant les fenêtres pour une fonction

spécifique, est également valable, vu que ces fenêtres pourront regrouper des formulaires et des icônes.

Finalement, comme nous l'avons déjà dit, le son peut être utilisé, notamment dans le sens "machine -> homme", pour des feed-backs et pour poser des questions à l'utilisateur.

Après avoir vu les différentes techniques d'interaction et celles a-priori plus adaptées aux applications interactives pour handicapés mentaux, introduisons un peu la problématique des utilisateurs.

### **1.2.3. La problématique des utilisateurs**

Dès 1971 Hansen donnait un certain nombre de principes de conception pour l'élaboration d'applications interactives graphiques. Nous les verrons en détail plus loin, mais disons que les 4 points principaux sont

- \* connaître l'utilisateur
- \* minimiser la mémorisation
- \* optimiser les opérations
- \* "engineer for errors".

Donc dans tout projet de ce type, il faut commencer par étudier l'utilisateur. L'ennui c'est que si tous les auteurs sont d'accord sur ce point, très peu expliquent comment "connaître les caractéristiques de l'utilisateur". Par exemple Hansen, comme nous le verrons, détaille ses trois derniers points, mais pas le premier. Par conséquent, un des objectifs de ce travail est aussi de montrer la démarche que nous avons suivie pour être "à l'écoute" des caractéristiques, des capacités et des besoins des utilisateurs. Comment nous avons essayé de travailler avec eux à chaque phase, et comment nous avons tenté d'exploiter au mieux les informations que nous obtenions (de façon directe ou indirecte ; via les psychologues et les éducateurs).

#### **1.2.3.1. Plusieurs interfaces pour les mêmes fonctionnalités**

Comme nous l'avons vu à la section 1.1.2., au niveau des utilisateurs souffrant d'un handicap mental, se pose le problème de la grande

hétérogénéité de la population. De cette hétérogénéité découle la nécessité de concevoir indépendamment les fonctionnalités de l'application interactive et son interface homme-machine, ceci afin de rendre possible la réalisation de plusieurs dialogues pour une même application. Le but est d'offrir différents dialogues susceptibles de correspondre à un grand nombre d'utilisateurs au sein de la population hétérogène. Nous reviendrons plus tard sur l'important principe de séparation modulaire entre les fonctionnalités et l'interface. En tous cas l'idée se profile de réaliser plusieurs interfaces adaptées à plusieurs "classes" d'utilisateurs. Plus exactement, nous avons voulu réaliser plusieurs éléments permettant le dialogue, qui peuvent être combinés pour essayer de rendre le dialogue le plus adéquat possible à un utilisateur (ceci sous-entend une possibilité de choix, individualisés, qui seraient faits, par exemple, par l'éducateur en fonction de l'utilisateur. Nous reviendrons sur cette idée de choix de paramètres).

#### 1.2.3.2. Une approche itérative

Pour élaborer ces différents dialogues (ou éléments de dialogue), nous avons décidé de suivre une méthode itérative. Elle consiste à faire participer les futurs utilisateurs à la définition, puis à l'évaluation de l'interface réalisée via une expérimentation, en vue de l'améliorer par des itérations successives.

Bien entendu cette méthode n'est pas sans inconvénient, car les itérations prennent du temps ; il faut implémenter, expérimenter, imaginer des modifications sur base des tests, puis les réaliser, etc.

Il existe une deuxième approche pour diminuer la lenteur des itérations et éviter de construire des versions d'interface inutiles. Cette méthode plus "théorique" consiste à se baser sur des modèles analytiques pour prévoir les performances de l'utilisateur avant de réaliser une version du dialogue.

L'inconvénient est que les modèles existants, sont, à notre point de vue, discutables. Comme tous les modèles, ils sont réducteurs et ils s'attachent souvent à un utilisateur relativement expert (et en tous cas pas handicapé mental).

### 1.2.3.3. Modèles de l'utilisateur et critiques

Citons tout de même un modèle qui présente l'homme en tant que processeur d'information. Il s'agit du modèle du processeur humain [Card, 1983]. L'homme y est modélisé en tant que système d'information ; ce système est divisé en 3 sous-systèmes interdépendants : les systèmes moteur, perceptif et cognitif. Chaque sous-système comprend un processeur et une mémoire. Ce modèle essaye de prédire les performances de l'homme au niveau de la perception, des capacités motrices, etc.

De ce modèle Card, Moran et Newel [1980] ont tiré le modèle "Goal, Operator, Method, Selection rule". Il analyse les tâches routinières exécutées par une personne. En fait, la structure cognitive de l'homme est représentée en terme de buts, d'opérateurs, de méthodes pour atteindre les buts et de règles de sélection permettant de choisir parmi plusieurs méthodes. Ce modèle essaye de prédire les méthodes que l'utilisateur choisira pour réaliser la tâche routinière et le temps qu'il lui faudra pour la réaliser.

Toutefois ce modèle souffre de sérieuses restrictions. En effet, il a été appliqué à l'édition de texte (qui n'est pas une tâche routinière pour tout le monde), mais surtout il suppose que l'utilisateur ne commet pas d'erreur et qu'il est un expert dans la réalisation de sa tâche. Or, si le but est de concevoir des interfaces adaptées aux utilisateurs, il faudra fournir les plus gros efforts pour les utilisateurs qui ne sont pas experts. Donc ces hypothèses émises par Card, Moran et Newel sont un frein à l'application du modèle dans le domaine de l'interface homme-machine, et la critique qu'en fait J.C. Ruche [Ruche, 1988] est suffisamment explicite : "ces hypothèses sont assez importantes quand on sait qu'il est plus difficile de concevoir une application interactive pour des novices et que des erreurs sont et seront toujours commises. De plus, le modèle s'applique à des applications de traitement de texte et ne propose pas de généralisation à d'autres types d'applications". Nous ajouterons, pour notre part, qu'il ne propose pas non plus de généralisation pour des utilisateurs ayant un handicap mental.

Par conséquent, dans le cadre du projet qui nous a occupé, nous avons opté pour une démarche empirique et itérative pour l'élaboration des interfaces utilisateurs. Nous nous sommes bien sûr basés sur les



connaissances qu'ont les psychologues et éducateurs de cette population, mais sans toutefois essayer de "modéliser" les utilisateurs.

Après avoir défini quelques termes, passé en revue les différentes techniques d'interaction et mis en évidence celles applicables pour des utilisateurs handicapés, nous avons introduit la problématique des utilisateurs et annoncé la démarche que nous avons employée tout au long du projet. Nous allons à présent voir quelques principes et règles qui sont valables pour la majorité des systèmes interactifs, y compris ceux que nous envisageons.

#### 1.2.4. Principes et règles

##### 1.2.4.1. La séparation modulaire entre l'interface et les fonctionnalités

Ce principe est indispensable vu l'hétérogénéité des utilisateurs et la nécessité de réaliser plusieurs interfaces pour les mêmes fonctionnalités, afin de rencontrer le plus possible les caractéristiques des utilisateurs. Cette séparation permet "l'indépendance de conception de l'interface par rapport à la conception des fonctionnalités de l'application, l'indépendance de l'implémentation de l'application face au matériel ou aux techniques d'interaction qui seront utilisés, et enfin la construction itérative de prototypes de l'interface" [Chadelon & Warnant, 1987]. Ce principe est repris par la plupart des chercheurs dans le domaine [Coutaz, 1986a, 1986b ; Konsynski, 1985 ; Rosenthal, 1983 ; Sufrin, 1986 ; Edmonds, 1981 ; Sproull, 1983 ; Draper, 1984 et Norman, 1984].

##### 1.2.4.2. Principes de Hansen

Dans cette section et la suivante seront présentés des principes de conception qui s'appliquent à la majorité des systèmes interactifs. Nous les reprendrons dans les chapitres suivants, pour voir comment ils auront été

plus ou moins respectés dans l'application que nous avons développée. Les principes de Hansen s'appliquent à l'élaboration d'applications interactives graphiques (ce qui sera le cas de la nôtre) ; comme nous l'avons introduit précédemment, ils sont au nombre de 4 :

- Connaître l'utilisateur (comme nous l'avons déjà dit, ce point n'est pas détaillé, contrairement aux autres)
- Minimiser la mémorisation
  - . permettre la sélection plutôt que l'entrée de données
  - . utiliser des noms plutôt que des nombres
  - . assurer un comportement prévisible de l'application
  - . permettre l'accès à des informations utiles
- Optimiser les opérations
  - . exécuter rapidement les opérations routinières
  - . modifier le moins possible l'affichage pendant l'exécution d'une opération (ceci n'exclut pas du tout les feed-backs)
- Engineer for errors
  - . produire de bons messages d'erreur
  - . concevoir l'application de manière à empêcher les erreurs les plus communes
  - . permettre des actions réversibles
  - . offrir de la redondance
  - . garantir l'intégrité des données en cas de panne software ou hardware.

Voyons maintenant ce que propose Ben Schneiderman [Schneiderman, 1987].

#### **1.2.4.3. Les 8 règles d'or de Ben Schneiderman**

- 1) viser l'uniformité : l'utilisation d'une application ne doit pas souffrir d'incohérences entre ses différentes interfaces. Par exemple, il faut veiller à garder des noms identiques pour des fonctions identiques (et bien sûr des noms différents pour des fonctions différentes)
- 2) permettre des raccourcis aux utilisateurs experts

- 3) offrir un feed-back informatif. Ce point sera largement développé dans les chapitres suivants. Dans notre application, nous avons misé sur une grande variété de feed-backs (avec de la redondance comme dans les principes de Hansen).
- 4) concevoir des dialogues permettant de découper la tâche de l'utilisateur, afin qu'il ne doive pas fournir un effort de concentration trop long avant de pouvoir clôturer une étape.
- 5) offrir une gestion simple des erreurs ; une erreur ne doit jamais entraîner des résultats catastrophiques et irréversibles (voir les principes de Hansen). L'utilisateur doit pouvoir corriger facilement. Il ne faut pas qu'il soit angoissé par le fait qu'une erreur pourrait causer des dommages irrécupérables.
- 6) permettre le retour en arrière ; un utilisateur peut à tout moment vouloir recommencer ou abandonner une partie de son travail, l'application doit le lui permettre.
- 7) offrir à l'utilisateur le sentiment de contrôler l'application ; en fait c'est l'utilisateur qui doit diriger l'application (même si parfois le contrôle peut être mixte, c'est-à-dire l'application prend le contrôle. Nous reparlerons de ce point dans les chapitres suivants). Il n'empêche que la plupart du temps c'est l'utilisateur qui doit avoir le contrôle du déroulement des opérations.
- 8) réduire la charge de mémorisation à court terme (voir les principes de Hansen).

Nous reprendrons donc ces règles et principes dans les chapitres suivants et nous verrons dans quelle mesure les interfaces construites répondent à ces critères généraux.

Voyons à présent les outils dont peut disposer un concepteur pour développer son application interactive et notamment la partie réalisant l'interface entre l'homme et la machine.

#### **1.2.5. Les outils disponibles**

Les outils existants peuvent être rangés dans deux catégories, d'une part les "boîtes à outils", et d'autre part les "systèmes génériques". Ceux que nous avons employés s'apparentent davantage à la première catégorie,

mais voyons brièvement ce qu'on entend par "boîte à outils" et "systèmes génériques".

#### 1.2.5.1 Les boîtes à outils

Il s'agit, en fait, d'une véritable collection de procédures offertes au concepteur pour réaliser ses interfaces homme-machine. Bien entendu ces procédures n'ont pas toutes le même niveau ; certaines sont de très "bas niveau" (se rapprochant du système d'exploitation), et d'autres permettent de mettre en oeuvre des techniques d'interaction (manipuler facilement les fenêtres, les menus, etc).

Les avantages des boîtes à outils sont :

- \* une très grande puissance ; il est possible avec toutes ces procédures de réaliser à peu près n'importe quoi
- \* l'extensibilité, puisque le concepteur peut écrire de nouvelles procédures (sans doute de plus haut niveau et qui utiliseront celles de plus bas niveau offerte par la boîte à outils "de base") et les ajouter à sa collection.

Les désavantages des boîtes à outils sont

- \* la complexité due au grand nombre de procédures offertes ; il est pratiquement impossible à une personne de connaître les spécifications de toutes ces procédures. De plus si des procédures de plus bas niveau permettent une très grande puissance, elles sont par contre plus complexes et demandent des efforts de compréhension plus grands.
- \* ensuite, une collection de procédures n'aide pas le concepteur dans sa décomposition modulaire (une boîte à outils n'offre pas d'aide méthodologique au concepteur).

#### 1.2.5.2. Les systèmes génériques

L'idée principale est que les systèmes génériques fournissent un

squelette standard sur lequel sont greffés les composants spécifiques aux applications (squelette et composants s'appuient sur les services d'une boîte à outils). On distingue deux types de systèmes génériques : les "applications extensibles" et les "systèmes de gestion du dialogue".

- \* une application extensible est constituée d'un squelette réutilisable qui doit être étendu pour répondre aux besoins de l'application à concevoir. Grâce à cette technique, toutes les applications réalisées seront très uniformes puisqu'elles partageront le même squelette de base (ce qui est assurément un avantage). Par contre, pour étendre le squelette de base, le concepteur doit connaître les fonctions offertes par la boîte à outils (ce qui nous ramène aux mêmes désavantages que pour les boîtes à outils).
- \* un système de gestion de dialogue veut proposer au concepteur un langage de haut niveau pour la conception de ses interfaces, "Il est composé de deux modules : un préprocesseur, qui reçoit en entrée les spécifications de l'interface et génère son implémentation, et le noyau d'exécution, qui arbitre les interactions entre l'utilisateur et l'application" [Ruche, 1988]. Toutefois les systèmes de gestion du dialogue existants sont souvent spécifiques à un type particulier d'interface (par exemple RAPID/USE est dédié à la création d'applications interactives d'interrogation de base de données).

Pour notre projet, nous avons utilisé une boîte à outils conventionnelle, offrant tout de même des procédures de haut niveau pour manipuler très facilement bon nombre d'objets interactifs (les impératifs économiques n'étant pas absents dans ce choix). Nous reparlerons plus loin de la boîte à outils que nous avons employée.

## Chapitre 2. Le contexte dans lequel le projet est né

### Introduction

Après avoir examiné assez théoriquement le domaine de l'informatique et du handicap et celui de l'interface homme-machine, ce chapitre 2 exposera comment est né ce projet, quels acteurs en étaient les initiateurs et quels acteurs sont venus s'y greffer. Nous verrons ensuite brièvement le logiciel réalisé sur Schneider PC 1512 Amstrad sur lequel nous nous sommes basés pour démarrer notre projet. Enfin, nous examinerons la machine que nous avons utilisée, à savoir le Commodore Amiga.

### 2.1. Les acteurs impliqués

#### 2.1.1. Le PSINHA

Au sein du Département de Psychologie de la Faculté de Médecine de Namur s'est créé, il y a quatre ans, un service de recherche, d'information et de documentation concernant l'informatique et les personnes avec un handicap. Le PSINHA (PSychologie-INformatique- HAndicap) est une équipe pluridisciplinaire composée de médecins, psychologues, informaticiens, logopèdes, ingénieurs, animateurs.

Le PSINHA propose une documentation (livres, revues, actes de colloques) portant d'une part sur le matériel utilisé et d'autre part sur des expériences d'utilisation de celui-ci par ou pour la personne avec un handicap, organise des groupes de travail et de réflexion, organise et participe à des colloques, propose des séances de sensibilisation et de formation aux possibilités d'utilisation de l'informatique par ou pour la personne handicapée au sein même de l'institution, etc.

#### 2.1.2. Les institutions demandeuses et participantes belges

Une des demandes ponctuelles qui sont parvenues au PSINHA émanait de deux institutions belges. Il s'agissait de concevoir et développer un logiciel d'aide à la gestion de budget pour des personnes handicapées

mentales adultes séjournant en institution. Le PSINHA a répondu à cet appel en créant un groupe de travail sous l'appellation "L'informatique comme outil de réadaptation fonctionnelle et sociale pour la personne handicapée". Ce groupe de travail regroupait non-seulement des membres du PSINHA, mais également des membres de diverses institutions désireuses de participer au projet.

La tâche du groupe de travail fut tout d'abord d'analyser l'existant, c'est-à-dire examiner comment la gestion de budget était réalisée dans les diverses institutions avant l'arrivée de l'informatique. Ces observations leur ont permis de décrire d'une part les fonctionnalités que devrait offrir un programme de gestion de budget pour personnes avec un handicap mental et d'autre part la forme du dialogue entre l'ordinateur et la personne handicapée. Le cahier de charge ainsi réalisé a alors été remis à l'analyste programmeur qui a développé le produit demandé sur un ordinateur Schneider PC 1512 Amstrad en Turbo Pascal. Celui-ci a alors été présenté au groupe de travail qui a critiqué le produit, proposé des modifications, des ajouts, ce qui a donné lieu à de nouvelles spécifications à développer. Cette façon de travailler constitue donc, comme nous l'avons indiqué dans le chapitre 1, un processus itératif de développement de programmes.

L'état final du produit ainsi obtenu sera exposé dans la section 2.2. Le manque de graphiques/pictogrammes a permis d'arriver à la conclusion qu'un autre type de machine que le PC était souhaitable pour arriver à des résultats satisfaisants, principalement au niveau de l'interface homme-machine. Le choix se porta alors sur le Commodore Amiga 2000 (pour le développement et l'Amiga 500 pour l'utilisation), machine très utilisée en Suisse dans le domaine des handicapés mentaux, ainsi qu'à l'Université Libre de Bruxelles (ULB).

### **2.1.3. La FST**

C'est à ce moment que nous avons commencé notre travail. Un autre acteur est alors intervenu et a pris également part à ce projet ; il s'agit de la Fondation Suisse pour les Téléthèses (FST).

La Suisse compte quelques 8000 para- et tétraplégiques, 7000 IMC, 10000 victimes de sclérose en plaques et 120000 rhumatisants souffrant

d'un mal aigu et déformant. Toutes ces personnes sont gravement atteintes dans leur mobilité. La FST, qui agit sur mandat, a pour objectif d'améliorer, dans toute la mesure du possible, leur condition de vie. Pour ce faire, elle recourt à des systèmes électroniques, baptisés "téléthèses", spécialement adaptés aux besoins et aux aptitudes intactes des handicapés. Ces appareils concourent à accroître l'autonomie de l'utilisateur en lui offrant un meilleur contrôle de son environnement.

Ainsi, l'objectif initial de la FST est l'aide aux handicapés physiques grâce à des appareils électroniques. On rappellera ici les deux principales réalisations de la FST dont nous avons parlé dans le chapitre précédent : HECTOR, la "machine qui parle" et JAMES, la télécommande infrarouge universelle. Depuis peu cependant, la FST a créé un nouveau service, appelé LOGIBABA, qui agit principalement dans le domaine "informatique et handicap mental".

#### **2.1.4. Les institutions participantes suisses**

Ce nouveau service est étroitement lié à La Castalie, institution pour handicapés mentaux à Monthey (Suisse), lieu de notre stage. Cette institution a participé de près à ce projet en nous mettant directement en contact d'une part avec des handicapés et d'autre part avec des psychologues et des éducateurs qui ont activement pris part aux phases de spécification et d'évaluation du programme.

D'autre part, n'étant en contact à La Castalie qu'avec des handicapés "scolarisés" de 16 à 23 ans, nous voulions être également en rapport avec des handicapés adultes exerçant d'une façon ou d'une autre un travail, ce qui a été possible grâce aux EPSE La Combe, les Etablissements Publics Sociaux-Educatifs de Genève, qui se sont à leur tour intégrés au projet.

#### **2.2. Le programme de gestion de budget réalisé sur PC**

Dans cette section, nous n'allons pas voir en détail comment se déroule ce programme. Comme nous l'avons dit, ce logiciel a servi de tremplin à notre travail. Il aurait été absurde de ne pas tenir compte de cette expérience et de repartir de zéro. Nous allons donc ici nous borner à présenter les éléments qu'il était, selon nous, impératif de garder dans



notre logiciel.

### 2.2.1. Sur le plan fonctionnel

Les spécifications des fonctionnalités indiquent que principalement trois possibilités sont offertes à l'utilisateur : spécifier une entrée d'argent (une recette), une dépense ou une économie.

Une entrée d'argent peut être de deux types : soit une entrée principale c'est-à-dire une somme que la personne reçoit chaque semaine le même jour (tous les lundis par exemple) ou une entrée secondaire ou non-principale. Lorsque l'utilisateur communique une entrée d'argent, il doit spécifier s'il s'agit d'une entrée principale ou secondaire.

Les dépenses constituent la grosse partie du budget proprement dit. La logique à suivre est la suivante : le premier jour de la semaine, la personne handicapée spécifie pour chaque jour quelles sont les dépenses qu'elle fait chaque semaine et qu'elle fera aussi cette semaine-ci (dépenses fixes). Le total de cette somme est directement retiré du total disponible, montrant ainsi au handicapé ce dont il peut disposer librement. Ensuite, chaque fois qu'une dépense est effectuée, on la communique au programme en spécifiant s'il s'agit d'une dépense fixe ou d'une dépense imprévue. Une dépense imprévue est automatiquement retirée de la somme disponible.

Enfin, ce programme permet à la personne handicapée de communiquer une certaine somme qu'elle désire économiser afin d'effectuer ultérieurement un achat plus important. Une première option permet au handicapé de se remémorer le total de ses économies, et de savoir, étant donné le montant de l'achat qu'il désire faire et qu'il spécifie, s'il a la possibilité d'acheter cet objet et, dans ce cas, combien il lui restera. Un deuxième choix donne à l'utilisateur la possibilité d'ajouter une certaine somme d'argent dans sa "tirelire". Enfin, la personne peut, grâce à la troisième option retirer effectivement de l'argent de ses économies afin de réaliser son achat.

On voit donc que ces trois possibilités constituent bien la gestion du budget. En effet, l'utilisateur va, en début de semaine, indiquer les dépenses qu'il a l'intention de faire et, jour après jour, il va communiquer

toutes les dépenses et les recettes qu'il a effectivement faites. De plus, le handicapé peut mettre de l'argent de côté et retirer de l'argent de ses économies tout comme il le fait dans la réalité.

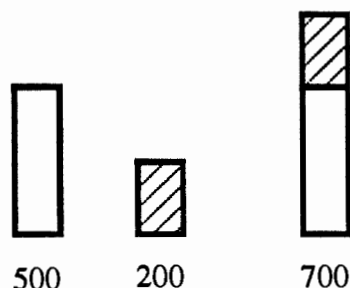
Il est clair que ces fonctionnalités doivent être gardées. Nous verrons dans le chapitre 4 qu'elles sont bien présentes dans nos spécifications, même si elles sont organisées de façon différente. On peut remarquer, dans le logiciel réalisé sur PC, que les deux notions de "tenue des comptes" et de "budget" sont groupées au sein de la même fonction d'enregistrement des dépenses. Nous avons préféré séparer ces deux activités pour diverses raisons dont l'une était qu'il nous fallait limiter la partie que nous allions implémenter si nous voulions arriver à un produit fini dans le temps qui nous était imparti. En effet, nous voulions mettre l'accent sur l'interface et essayer de trouver des dialogues qui soient les mieux adaptés possibles aux utilisateurs handicapés ; pour ce faire, il nous fallait relativement beaucoup de temps.

### 2.2.2. Sur le plan de l'interface

Le Schneider PC 1512 n'est pas une machine conçue pour une gestion aisée des graphiques et du son, ce qui explique qu'il a fallu se limiter à une interface principalement alphanumérique, c'est-à-dire que les informations destinées à l'utilisateur apparaissent à l'écran sous la forme de suites de caractères et que les données nécessaires au programme sont saisies via le clavier.

Un certain niveau graphique a néanmoins été implémenté. Un "thermomètre" de couleur tente de faire acquérir à la personne handicapée mentale la notion de grandeur numérique, les relations de comparaison numérique "est plus grand que", "est plus petit que", "est égal à" et les notions d'addition et de soustraction.

Ainsi, on trouvera des graphiques tels que celui-ci :



où, ayant une somme de 500 francs, je lui ajoute une somme de 200 francs (200 est donc inférieur à 500 puisque la colonne représentant 200 est plus petite que celle représentant 500) pour obtenir un montant de 700 (700 est donc supérieur à 500 et à 200 et de plus,  $500 + 200 = 700$  puisque si l'on place la colonne du 200 sur la colonne du 500, on obtient la colonne représentant 700).

De plus, une réglette graphique représente le temps. Ceci permet à la personne handicapée de mieux se situer dans le temps, de comprendre la succession des jours dans la semaine. Cette réglette se présente de la façon suivante :

MER	15
JEU	0
VEN	55
SAM	0
DIM	25
LUN	13
MAR	0

Il faut noter que la semaine ainsi représentée ne commence pas par le lundi. Ceci est dû au fait que l'entrée principale pour cet utilisateur a lieu tous les mercredis. Cela permet au handicapé d'avoir ainsi un point de repère stable, ce qui est également le cas pour les personnes non-handicapées pour lesquelles le jour repère est le lundi, ce qui correspond au premier jour de travail. Pour beaucoup de handicapés vivant constamment en institution, la notion de week-end n'existant pas, ce

repère ne peut pas être utilisé.

Une croix barrant une case montre à l'utilisateur le jour actuel de la semaine.

Les nombres indiqués dans les cases représentent les prévisions des dépenses fixes déterminées par le handicapé le premier jour de sa semaine. Ils sont à 0 tant que l'utilisateur n'a pas spécifié ses dépenses fixes.

Il est clair que ces réalisations graphiques peuvent aider les handicapés mentaux à mieux comprendre les notions qui y sont associées. En ayant une machine conçue davantage pour la construction d'interfaces graphiques, on comprend aisément que c'est dans cette voie qu'il faudra approfondir, les fonctionnalités étant quant à elles bien spécifiées.

Il faut bien se rendre compte, lorsque l'on développe des programmes pour handicapés mentaux qui seront principalement utilisés dans des institutions, que des contraintes parfois contradictoires pèsent sur les initiateurs de ce type de projets. Il est clair que le groupe de travail était tout à fait conscient de la nécessité de graphiques et de pictogrammes afin que le logiciel puisse être une aide valable pour les handicapés. Dès lors, ils auraient très bien pu se décider à développer sur un MacIntosh de type Mac II dont le niveau graphique est très élevé. Cependant, il faut bien comprendre que les institutions vont devoir également acquérir ce matériel afin de pouvoir exploiter le logiciel. Or, généralement, celles-ci ne disposent que d'un budget limité pour le matériel informatique et il faut se rendre à l'évidence que l'achat d'un Mac II constitue un investissement trop important. C'est pourquoi il avait été décidé de développer sur un PC en attendant de trouver une machine plus intéressante au prix raisonnable. Ce fut le Commodore Amiga que nous présenterons à la section suivante.

Cette contrainte budgétaire ne se pose pas que pour la machine. En effet, lorsqu'on recherche le moyen de saisie le mieux adapté, on peut envisager toutes sortes de solutions : clavier, souris, écran tactile, crayon optique, table graphique, système de reconnaissance de la parole, ... Ainsi, dans le domaine des handicapés mentaux, il est plaisant d'imaginer un programme où l'utilisateur sélectionne les objets interactifs en posant son doigt sur l'écran à l'endroit où est situé cet objet. Mais bien sûr, le prix

d'un tel écran tactile est très élevé et il faut voir si l'on peut se permettre cet achat.

Dans le chapitre 1, nous avons souligné la nécessité d'une connaissance approfondie des utilisateurs afin de pouvoir imaginer l'interface la meilleure qui soit. Mais l'imagination peut être très fertile et envisager des moyens très attrayants, mais dont le prix est inaccessible. Il faut donc également toujours garder en mémoire cette contrainte budgétaire, car à quoi cela sert-il de développer un programme ultra- adapté aux utilisateurs si le prix d'exploitation en rend l'acquisition impossible ?

### **2.3. Le Commodore Amiga**

Comme nous l'avons dit plus haut, la pauvreté de l'interface qu'il est possible de construire sur un PC (dans un temps acceptable) ont poussé à acquérir une autre machine. Le choix s'est ainsi porté sur le Commodore Amiga. Cette décision s'est essentiellement basée sur deux caractéristiques propres à l'Amiga qui sont les suivantes. Voyons brièvement ces caractéristiques techniques.

#### **2.3.1. La gamme Amiga**

La première caractéristique intéressante est que Commodore propose deux machines dans sa gamme Amiga : le 2000 et le 500. La différence entre ces deux machines a trait principalement au nombre de connecteurs d'extension (extension de mémoire centrale, extension par micro-processeurs, extension pour unités de disquettes, connecteurs compatibles PC-XT, PC-AT) et aux options possibles (compatibilité par carte passerelle PC-XT, PC-AT, MacIntosh, Interface MIDI pour exploiter au maximum les capacités sonores, Genlock pour pouvoir ajouter des graphiques sur des bandes vidéo,...). Ces différences ont tendance à faire de l'Amiga 2000 une machine de développement, alors que le 500 constitue davantage une machine d'exploitation, d'utilisation. Ces deux ordinateurs sont totalement compatibles (ils ont quasi la même architecture), mais leur grande différence de prix a été prépondérante dans la prise de décision. En effet, nous avons vu dans le point 2.2.2.

qu'une règle importante à respecter était de toujours garder en mémoire que les institutions pour handicapés ne disposent pas d'un budget énorme à allouer à l'informatique. L'Amiga permet donc de respecter cette règle en développant sur Amiga 2000 et en exploitant le logiciel obtenu sur Amiga 500, ce qui ne constitue pas un investissement très élevé pour les institutions.

### **2.3.2. Les capacités graphiques de l'Amiga**

La caractéristique la plus importante a trait aux capacités graphiques et sonores de l'Amiga. Que l'ordinateur soit capable de calculer d'importantes quantités de feuilles de salaire par seconde ou qu'il soit plus rapide qu'un vieux calculateur ne sont certainement pas des critères prépondérants pour l'achat d'un Amiga. On ne choisit pas un Amiga parce qu'il est rapide ; en effet, sa fréquence d'horloge n'est que de 7,14 MHz et ses accès aux disquettes sont relativement lents. Par contre, avoir la possibilité d'afficher et de traiter des images d'une très bonne qualité tout en écoutant de la musique peut attirer l'attention quand on veut développer des applications interactives pour handicapés mentaux.

#### **2.3.2.1. Caractéristiques générales de l'Amiga**

##### **1) Graphiques couleurs sophistiqués**

Les possibilités graphiques avancées de l'Amiga permettent de réaliser des écrans couleurs haute résolution, en mode statique ou animé. La résolution normale d'un écran est de 320 x 200 ou 320 x 400 pixels. La haute résolution correspond à 640 x 200 ou 640 x 400 pixels. Selon le mode graphique, on peut sélectionner 16 ou 32 couleurs affichables en même temps, choisies dans une palette de 4096 couleurs.

##### **2) Son stéréo digitalisé**

L'Amiga a des possibilités étendues en stéréophonie à quatre voix réparties sur les deux canaux. Diverses options (telle l'interface MIDI) peuvent être ajoutées pour profiter mieux encore des possibilités sophistiquées de sonorisation et de musique stéréophoniques digitalisées de l'Amiga.

### 3) Interface MacIntosh

l'interface du système ressemble étrangement à celle du MacIntosh. Ainsi, on trouvera sur l'écran des icônes représentant les disquettes, le disque dur, les différents fichiers et la poubelle, ainsi que des menus déroulants. La souris est évidemment présente et comporte deux boutons, celui de gauche pour sélectionner (1 clic) ou exécuter un fichier ou ouvrir une directory (2 clics), et celui de droite pour sélectionner des options dans les menus déroulants.

### 3) Traitement multitâches

Le traitement multitâches, c'est-à-dire la gestion simultanée de plusieurs tâches, est une possibilité que l'Amiga est le seul à offrir parmi les micro-ordinateurs de sa catégorie de prix ou des catégories voisines. Par exemple, il est possible de faire tourner en même temps un programme de calcul et un programme de traitement de texte. En cas de compétition entre les différents programmes pour l'utilisation des ressources de l'ordinateur, celui-ci alloue les ressources en fonction des priorités qu'on lui aura spécifiées.

### 4) Conversion directe texte/parole

Les mots tapés au clavier peuvent être convertis directement en parole par l'utilisation du synthétiseur de parole SAY. Cette possibilité est inédite, mais elle n'est pour l'instant malheureusement disponible que dans une version anglophone.

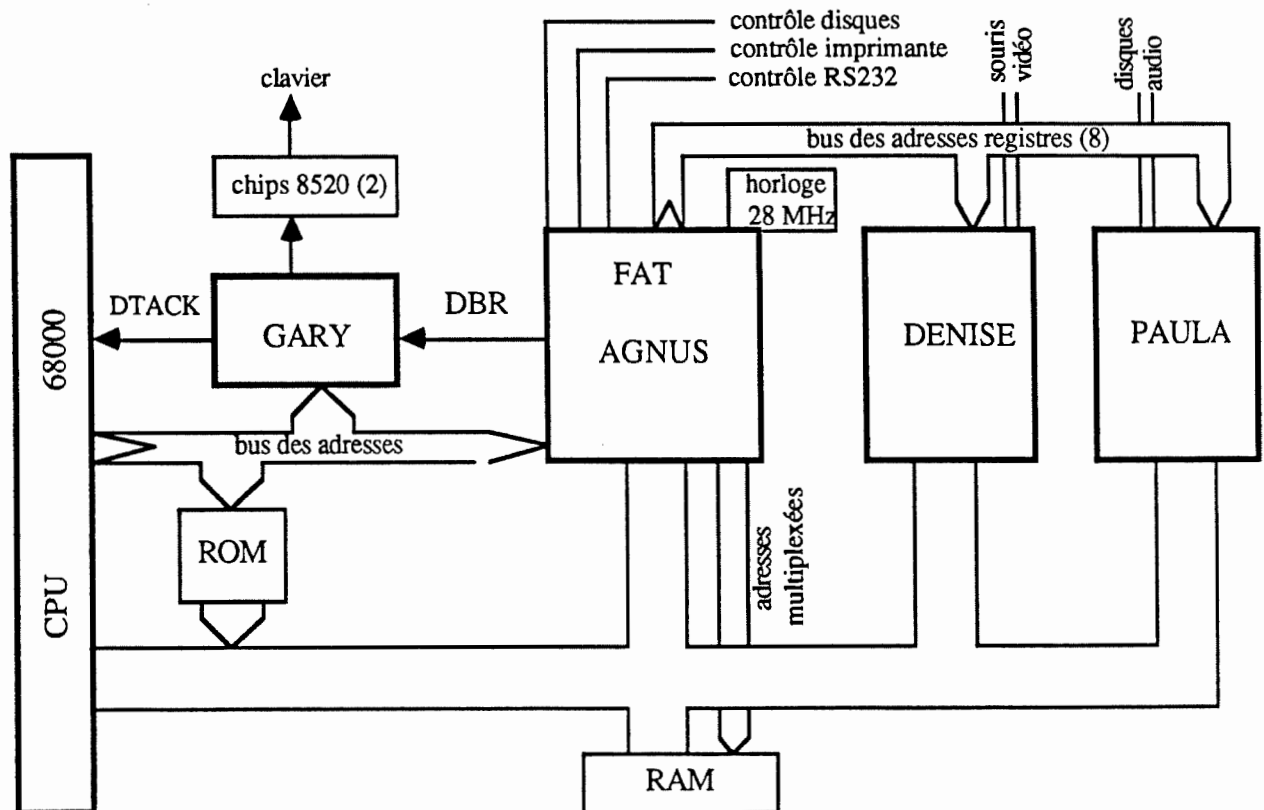
### 5) Format standard de fichiers

Les micro-ordinateurs Amiga utilisent un format spécial de fichiers appelé *Interchange File Format* (format pour l'échange de données) ou tout simplement IFF. Ce format permet d'avoir une structure standard des fichiers Amiga, quelque soit leur contenu : code de programme, données graphiques ou sonores. Avec le format IFF, l'Amiga peut échanger des données entre différents types de programme.

### 2.3.2.2. Schéma fonctionnel de l'Amiga

Le but de cette section est d'éclairer le hardware de l'Amiga qui permet ces fantastiques possibilités graphiques et sonores. Nous ne verrons pas en profondeur l'architecture de l'Amiga, mais nous nous en tiendrons à ce que nous jugeons nécessaire à la compréhension du fonctionnement en général.

Les éléments de base permettant les possibilités citées ci-dessus se réduisent à trois circuits. Ils ont été conçus pour l'Amiga et portent le nom de *circuits spécialisés* (on appelle circuit spécialisé, un circuit intégré, conçu par une firme spécialisée dans les semi-conducteurs, pour une machine déterminée, dans un but précis). Leur dénomination sont 8370, 8362 et 8364. Les concepteurs les baptisèrent respectivement FAT-AGNUS (version ultérieure plus puissante de l'ancien AGNUS), DENISE et PAULA. On trouve également le processeur Motorola 68000 qui constitue l'unité centrale du système, deux circuits d'entrée/sortie parallèles qui sont des circuits périphériques 8520 et un composant de contrôle baptisé GARY .





Le 68000 communique avec le reste de l'ordinateur par l'intermédiaire d'un bus d'adresses, d'un bus de données et de lignes de commande. Le schéma ci-dessus est bien évidemment très incomplet (principalement en ce qui concerne les lignes de commande). La commande du bus de données peut être prise par le 68000 ou par FAT-AGNUS. Lorsque FAT-AGNUS prend le contrôle du bus, il envoie un message à GARY, appelé DBR et indiquant qu'il a le contrôle. Si le 68000 désire y accéder, il doit vérifier qu'il a bien reçu le signal DTACK de la part de GARY qui lui annonce que le bus de données est libre et qu'il peut lâcher ses données sur le bus.

La tâche de GARY est de :

- engendrer tous les signaux de commande des bus
- assurer tous les décodages d'adresses
- traiter une partie des circuits de disquette
- assurer l'interface de réinitialisation du clavier

Comme on peut le voir sur le schéma, GARY est connecté à deux circuits multifonctionnels 8520 d'entrée/sortie qui ont les attributions suivantes :

- entrées/sorties par le port parallèle
- lignes de commande vers et depuis les ports de la souris et des manettes de jeu
- une ligne de commande vers les indicateurs lumineux de la face avant
- les lignes internes de commande
- les lignes de commande du clavier, l'horloge et les données
- les lignes de commande du port série
- les lignes de commande de l'interface de l'unité de disquette
- les rythmeurs internes

Les trois circuits spécialisés ont des registres de commande, ce qui explique la présence du bus des adresses registres. Ces bus sont généralement chargés par le 68000, mais FAT-AGNUS est aussi en mesure de charger des registres de commande dans les deux autres circuits spécialisés. Il faut noter également que ces circuits ne sont pas

indépendants ; en effet, pour la gestion des graphiques par exemple, ce n'est pas un seul circuit qui s'en occupe mais bien deux des trois circuits.

FAT-AGNUS remplit les fonctions suivantes :

- permet l'animation à grande vitesse
- engendre les signaux de commande pour la mémoire RAM et pour les extensions de mémoire
- engendre les adresses multiplexées pour la mémoire RAM et pour les extensions de mémoire
- engendre toutes les horloges du système à partir de l'oscillateur à 28 MHz.

Denise, quant à elle, a pour rôle de générer les images, à partir de données provenant de la RAM ; Denise renferme également le compteur de la souris, c'est-à-dire les coordonnées X-Y de la souris.

Enfin, Paula gère les traitements des entrées/sorties, disquettes et les sons. De plus, Paula sert de support à la gestion des interruptions. Toutes les interruptions présentes dans le système passent par elle.

Ainsi, ces trois circuits prennent en charge la création des sons, la reproduction des images et les accès disquettes indépendamment du processeur. Ceci implique donc que l'on peut à la fois afficher des images tout en écoutant de la musique sans que cela gêne beaucoup le processeur 68000 qui peut continuer à travailler.

### 2.3.3. Les configurations utilisées dans le projet

Pour le développement , nous avons donc utilisé un Amiga 2000 (1MB de mémoire centrale) ayant un lecteur de disquettes 3 1/2 pouces, un disque dur de 20 MB, un écran couleur et une extension de mémoire de 3MB.

Pour l'exploitation en institution, il est nécessaire de disposer d'un Amiga 500 (512 KB) ayant DEUX lecteurs de disquette 3 1/2 pouces (1 en option ; nécessaire sinon, l'utilisation nécessitera trop de swaps disquettes)

et une extension de mémoire à 1MB (nécessaire car le programme a besoin de l'horloge interne qui se trouve, pour l'Amiga 500, sur la carte d'extension de mémoire).

## Chapitre 3. La démarche de spécification

### Introduction

Ce chapitre présente la première phase du cycle de vie du projet, la spécification. Nous verrons que dans la démarche suivie, les fonctionnalités et les interfaces sont spécifiées parallèlement, car si l'indépendance modulaire entre ces deux parties est toujours souhaitable à la conception, elle l'est beaucoup moins à la spécification. Nous verrons comment les utilisateurs entrent en jeu dès cette phase, et le rôle crucial des psychologues et des éducateurs. Le processus est itératif et nous passerons en revue quelques unes des "itérations" que nous avons vécues durant cette phase.

### 3.1. Remarques méthodologiques

#### 3.1.1. Les 4 idées de base

Cette section reprend quelques thèmes importants présentés dans le chapitre 1, qui traitait de l'informatique et handicap et de l'interface homme-machine en général. Quelle que soit la démarche suivie pour concevoir une application interactive, les 4 points suivants sont réalisés (suivant la méthode, l'accent sera mis sur l'un ou l'autre) :

- \* analyser les fonctionnalités que doit réaliser l'application
- \* connaître les utilisateurs concernés
- \* étudier les interfaces possibles
- \* suivre une approche itérative.

#### 3.1.1.1. Analyser les fonctionnalités

En effet, le but premier d'une application est de réaliser, d'exécuter exactement les fonctionnalités désirées (ce qui est vrai pour les applications en temps différé, reste vrai pour les applications interactives). Il ne sert à rien de développer une application avec une interface

parfaitement adaptée aux utilisateurs, si elle ne réalise pas ce que l'utilisateur attend d'elle. Ce point semble évident, mais il faut toujours l'avoir en tête et ne pas l'oublier sous prétexte que le but est avant tout de concevoir des dialogues adaptés.

#### **3.1.1.2. Connaître les utilisateurs**

Nous avons déjà parlé à la section 1.1.2. de l'hétérogénéité de la population handicapée mentale et dans la section 1.2.3. nous avons vu qu'il en découle le besoin de concevoir plusieurs interfaces pour les mêmes fonctionnalités. Dans notre projet, nous avons voulu mettre l'accent sur cette connaissance des utilisateurs. Nous verrons comment ils ont participé à la plupart des phases et comment nous avons collaboré avec les éducateurs et les psychologues qui connaissent le mieux les handicapés mentaux.

#### **3.1.1.3. Etudier les interfaces**

Ce point se base fortement sur les 2 précédents. les dialogues doivent permettre aux utilisateurs d'avoir accès aux fonctionnalités offertes par l'application. Par conséquent pour réaliser des dialogues qui auront une chance d'être adaptés, il faut prendre en considération les caractéristiques des utilisateurs pour lesquels ils sont destinés.

#### **3.1.1.4. Suivre une approche itérative**

Nous avons vu au point 1.2.3.3. que les modèles de l'utilisateur sont peu nombreux et criticables, qu'ils sont difficilement applicables pour concevoir une application interactive adaptée (et encore moins dans le cas d'utilisateurs handicapés mentaux). Par conséquent, il est clair qu'il est impossible de réaliser une application interactive parfaite à la première tentative. Si on peut raisonnablement supposer que les fonctionnalités seront directement bonnes, les dialogues devront sans aucun doute faire l'objet de modifications, d'adaptations, voire de refontes complètes. L'approche itérative a été largement employée au cours des différentes

étapes du cycle de vie du projet, et notamment celle des spécifications (comme nous le verrons dans ce chapitre).

### 3.1.2. Une méthodologie de spécification d'une interface homme-machine

Nous voudrions dans cette section, présenter quelques éléments importants du travail d'élaboration d'une méthodologie qu'a réalisé J.C. Ruche [Ruche, 1988], nous verrons ensuite dans quelle mesure il nous a été possible de suivre cette voie.

#### 3.1.2.1. Distinction entre tâche et présentation

Dans la méthodologie proposée, la spécification de l'interface se fait en 2 parties ; d'une part la spécification de la Tâche, et d'autre part celle de la Présentation. Voyons ce que l'on entend par tâche et présentation.

En fait une tâche est définie comme un ensemble de messages interactifs sur lesquels l'utilisateur peut effectuer des opérations, des contraintes d'enchaînement pouvant être exprimées sur ces opérations. Un message interactif peut être de saisie ou d'affichage. Il faut aussi faire la distinction entre les messages interactifs fonctionnels (qui contiennent des informations permettant l'accomplissement de la tâche de l'utilisateur) et les messages purement interactifs (qui aident l'utilisateur mais n'ont aucune influence sur les fonctionnalités).

Donc l'utilisateur effectuera la tâche (pour laquelle l'application interactive a été développée) en effectuant des opérations sur les messages interactifs qui lui sont présentés. Mais il faut bien comprendre que les messages interactifs ne font aucune hypothèse sur la présentation qui en sera faite.

La Présentation "traduit" les messages interactifs en objets interactifs ; les objets sont la visualisation à l'écran des messages interactifs (en réalité il ne s'agira pas toujours de visualiser à l'écran ; on peut par exemple prononcer un message à l'aide d'une synthèse vocale, c'est aussi une présentation de message).

En toute généralité, un objet interactif présentera un ou plusieurs

messages interactifs, ou même un seul message interactif sera présenté par un ou plusieurs objets interactifs (nous en reparlerons avec les billets de banques).

Les objets interactifs utiliseront les techniques d'interaction relevées à la section 1.2.2.

### **3.1.2.2. Spécification de l'interface scindée en deux parties**

La division de l'interface en deux parties, la tâche et la présentation, a amené l'auteur à poser l'hypothèse suivante : la spécification de la tâche doit être indépendante de la présentation.

L'analyse de la tâche se fait indépendamment de toute implémentation puisqu'elle porte sur les informations qui seront communiquées à l'utilisateur (les messages interactifs) mais pas sur leurs présentations.

Donc, dans cette méthodologie, on commence par spécifier les différents messages interactifs, les opérations qu'il est permis de réaliser sur ces messages et cela sans parler des présentations possibles.

La spécification de la présentation est donc la définition d'objets interactifs traduisant les messages interactifs. Le choix de ces objets (rappelons qu'il ne s'agit pas uniquement d'objets affichés à l'écran) est critique, puisqu'ils sont ce que l'utilisateur verra ou entendra de l'application interactive.

Par conséquent nous y avons attaché une grande importance, et c'est surtout à ce niveau qu'ont eu lieu les itérations de la phase de spécification.

Il faut ajouter enfin que la spécification de la présentation se fera avec la connaissance du matériel utilisé (hardware ou software).

### **3.1.2.3. Applicabilité de la méthodologie dans notre projet**

En fait dans cette méthodologie, la tâche constitue, en quelque sorte, la définition abstraite de l'interface, la présentation étant la concrétisation de la tâche. Nous verrons que dans notre cas, la dichotomie abstrait-concret n'est pas toujours si simple (il y aura un troisième niveau, ou bien non pas une abstraction mais plusieurs ; nous y reviendrons).

De toute façon, l'avantage de cette méthodologie est de voir qu'il y a d'une part l'interface plus abstraite (la tâche), en terme de messages interactifs et d'opérations possibles, et d'autre part la concrétisation de la tâche (la présentation). Cette tâche peut être étudiée avec plus ou moins de précision, sans poser la moindre hypothèse au niveau de la représentation des messages à l'utilisateur, et peut utiliser un processus itératif pour spécifier les présentations possibles (ce qui nécessite la connaissance du matériel et autant que possible de l'utilisateur).

Comme nous l'avons dit, nous n'avons pas toujours pu appliquer la méthodologie proposée et notamment la dichotomie entre tâche et présentation. Nous avons travaillé de façon plus pragmatique et il serait faux de dire que la tâche était complètement et rigoureusement spécifiée avant de nous préoccuper de la spécification de la présentation. Comme nous le verrons dans la section suivante, ceci est dû, en partie, au fait que nous ne savions pas avec précision ce que nous pouvions demander de faire aux utilisateurs.

### **3.1.3. Les spécifications des fonctionnalités et de l'interface en parallèle**

Dans la section précédente nous avons parlé de la spécification de l'interface, celle-ci pouvant être scindée en 2 parties, la spécification de la tâche et celle de la présentation. Nous n'avons pas parlé de la spécification des fonctionnalités.

Dans une vision traditionnelle, on serait tenté de dire que les fonctionnalités doivent être entièrement spécifiées avant de commencer à spécifier l'interface. Ceci découle de la tendance plus ancienne qui consiste à partir des fonctionnalités d'une application interactive pour ensuite venir ajouter l'interface utilisateur, qui est alors considérée comme un module supplémentaire.

Nous avons préféré suivre une démarche qui centre la conception sur l'utilisateur. Les fonctionnalités de l'application doivent être comme une



panoplie de services rendus à l'utilisateur, et elles lui sont accessibles via l'interface. En fait nous ne désirons pas que ces 2 spécifications soient faites indépendamment. Dans l'évaluation de sa méthodologie, qui supposait de terminer la spécification des fonctionnalités avant celle de l'interface, J.C. Ruche disait "Le fait de débiter par la spécification de l'application proprement dite a une incidence sur la qualité des informations qui seront offertes à l'utilisateur car elles ont été définies sans tenir réellement compte des besoins de ce dernier. (...) Cette approche semble dès lors contraignante puisqu'elle ne centre pas suffisamment la spécification des fonctions sur l'utilisateur." [Ruche, 1988].

Par conséquent, nous avons décidé de réaliser les spécifications des fonctionnalités et des interfaces plus ou moins en parallèle et de manière itérative. Donc les spécifications des dialogues pourront, au cours de leur élaboration, provoquer des modifications dans les spécifications des fonctionnalités. Il n'y a pas d'indépendance entre ces deux spécifications puisque les informations que s'échangent les fonctionnalités et l'interface proviendront autant des besoins de l'utilisateur que de ceux des fonctions de l'application.

Toutefois ceci ne remet pas en cause le principe de séparation modulaire entre les fonctionnalités et l'interface évoqué à la section 1.2.4.1. Ce principe vaut pour la conception et non pas la spécification, et l'indépendance, entre les modules de l'interface et ceux qui réalisent les fonctionnalités, qui en découle, reste le principe de base sur lequel doit être construite toute application interactive. C'est par cette indépendance qu'il sera possible de concevoir plusieurs interfaces pour un même jeu de fonctionnalités.

Il reste à ajouter que si méthodologiquement il était souhaitable de mener de front le travail de spécification des fonctionnalités et celui de spécification des interfaces, cela était préférable aussi pour nous, car n'ayant jamais été en contact avec des personnes handicapées mentales avant ce projet, il nous était difficile d'imaginer ce que l'on pourrait attendre d'elles. Donc, dès le départ, nous avons décidé de ne rien spécifier sans d'abord entrer en contact avec les futurs utilisateurs, mais aussi les gens qui les connaissent le mieux.

Ces rapports avec les futurs utilisateurs et avec les psychologues font

l'objet de la section suivante.

### **3.2. La participation des utilisateurs**

Comme nous l'avons souligné à la section 1.1.3.6., le développement de logiciels pour handicapés mentaux nécessite une approche spécifique. Dans le cas d'applications standard, les concepteurs peuvent se baser sur leur expérience personnelle, mais pour des personnes handicapées mentales, il est nécessaire de connaître, le mieux possible, les capacités des futurs utilisateurs. Pour ce faire, il est indispensable de bénéficier de l'expérience qu'ont les éducateurs et les psychologues. Mais nous verrons que les utilisateurs doivent eux aussi collaborer dans le travail de spécification (notamment de l'interface). Voyons quels sont les problèmes et comment les résoudre.

#### **3.2.1. Un problème de communication**

Dans le cadre d'applications interactives standard, les futurs utilisateurs peuvent être impliqués dans la définition des besoins et dans les spécifications des fonctionnalités (et de l'interface) qu'ils souhaitent voir réalisées par l'application. Cette participation est possible mais elle ne va pas sans heurts ; les utilisateurs changent d'avis, les concepteurs n'écoutent pas toujours les demandes émises... En fait, l'important est d'être conscient que les spécifications vont inmanquablement "glisser", et qu'il est vain d'espérer avoir directement les spécifications finales.

Il n'empêche que, moyennant quelques concessions de part et d'autre, la collaboration des futurs utilisateurs d'une application à ses spécifications est possible et souhaitable.

Dans le cas d'utilisateurs handicapés mentaux, s'il est toujours souhaitable qu'ils participent, c'est par contre beaucoup plus difficile. Le problème est avant tout lié à la communication ; les futurs utilisateurs ne vont pas dire d'eux-mêmes ce qu'ils veulent que l'application réalise, il ne peut leur être demandé de remettre un rapport sur leurs besoins. Donc, si

dans le cadre d'applications classiques, le dialogue entre concepteurs et utilisateurs sera (ou devrait être) équilibré, ici par contre il est plus difficile de faire s'exprimer les utilisateurs. Pourtant, savoir ce qu'ils veulent ou peuvent réaliser avec une application interactive, et savoir quels types d'interfaces leurs conviennent, est d'une importance capitale (même plus que pour les applications classiques, où finalement on peut s'attendre aux besoins des utilisateurs et au genre de dialogue qu'ils désirent).

Par conséquent, il va falloir opérer d'une manière différente pour obtenir les informations que donnent habituellement les utilisateurs.

### 3.2.2. Le rôle des psychologues

#### 3.2.2.1. Psychologues actifs, concepteurs passifs

Il faut être conscient que notre rôle pendant la première partie de cette phase de spécification (des fonctionnalités et de l'interface) doit être passif.

Ce sont les psychologues et les éducateurs qui connaissent le mieux le problème. Normalement, ce sont eux les demandeurs du projet, et non pas les utilisateurs comme cela peut être le cas pour des applications classiques. Donc ils vont exprimer des besoins, ils vont dire ce qu'ils veulent voir réaliser par l'application, et surtout ils vont donner leur expérience de l'éducation spécialisée et des handicapés mentaux. Il s'agit donc d'une phase "d'écoute" pour ceux qui réaliseront l'application.

#### 3.2.2.2. Remarques sur notre passivité

Toutefois, nous n'avons pas été entièrement passifs, pour deux raisons : premièrement, s'il est vrai que nous n'avons jamais été en contact avec la population handicapée, nous ne partions pas dans le vide au moment d'aborder cette phase puisque nous avons bénéficié du travail réalisé par le PSINHA et le programme que nous avons présenté au chapitre 2 ; deuxièmement, nous devons nous préoccuper de la "faisabilité" du point de vue informatique : il n'est pas possible de réaliser toutes les

demandes exprimées par les psychologues. Donc dès le départ, et en connaissance du matériel utilisé (puisque finalement l'application devra être réalisée à l'aide de hardware et de software déterminés), il est nécessaire de s'assurer qu'on reste dans les limites du faisable.

### **3.2.2.3. La collaboration souhaitée**

Donc, en toute généralité, cette première partie de la phase de spécification doit voir la collaboration entre les informaticiens et les psychologues. Sur base de leur expérience et de leur connaissance, ces derniers exprimeront les besoins, les caractéristiques et les capacités des futurs utilisateurs. Commence alors le processus itératif : comme nous le verrons, les premières idées au niveau des fonctionnalités souhaitées et des modes de dialogue sont exprimées. Il faut s'assurer que tout est faisable a-priori avec le matériel à disposition. Ensuite ces spécifications s'affineront au fil des itérations (qui feront intervenir des tests auprès des utilisateurs, voir section suivante).

On voit que ce travail demande une approche pluridisciplinaire, où principalement deux disciplines collaborent, à savoir l'informatique et la psychologie. Cela ne va pas sans heurts puisque tous ne poursuivent pas nécessairement les mêmes objectifs, et surtout il n'est pas toujours sûr que les deux parties se comprennent, comprennent les objectifs de l'autre. Toutefois, on peut dire que cette collaboration est tout à fait possible, et en tout cas nécessaire, si psychologues et informaticiens se donnent pour but de satisfaire les utilisateurs finals.

### **3.2.3. Le rôle des utilisateurs**

Il n'empêche que s'il est nécessaire de travailler en collaboration avec les psychologues et les éducateurs, il est tout aussi important de travailler avec les utilisateurs, qui sont les personnes à qui est destiné le projet, et qui doit leur être adapté autant que faire se peut. Comme nous l'avons vu, une approche spécifique est souhaitée. Voyons concrètement ce qu'il est possible de faire.

### 3.2.3.1. Prise de contact

Il peut sembler inutile de nous attarder sur ce point, pourtant cela nous paraît nécessaire. En effet, on peut lire autant de rapports que l'on veut, parler avec les éducateurs et les psychologues, cela ne remplace pas les choses que l'on peut apprendre par contact direct avec les personnes. Il était pour nous hors de question de faire un quelconque travail sans vivre un peu avec les personnes à qui il allait être destiné. Par conséquent, comme nous le verrons en détail à la section 3.4., nous pensons qu'il faut passer du temps avec les handicapés mentaux, si on veut espérer faire quelque chose qui s'adapte à leurs besoins et à leurs caractéristiques. Donc, c'est encore un rôle passif que nous avons joué pendant cette première phase. Nous avons fait un certain nombre d'activités (éducatives ou non) avec quelques jeunes, et nous sommes ainsi entrés dans ce monde particulier.

Bien entendu cette prise de contact est d'autant plus importante si c'est la première avec la population handicapée (ce qui était notre cas).

### 3.2.3.2. La collaboration possible

Par la suite, il sera possible de passer à une phase de collaboration où les concepteurs sont plus actifs ; nous entendons par là le processus itératif auquel tous les acteurs participent : psychologues et informaticiens imaginent des tâches à faire faire aux utilisateurs, des dialogues, ensuite des petits tests sont réalisés avec ces utilisateurs, permettant de corriger, de modifier ou d'abandonner certaines idées. Nous reviendrons sur le genre de tests que nous avons faits et que nous jugeons possibles. L'important est que nous posions la démarche possible et l'idée d'une collaboration entre les représentants de la psychologie ou de la pédagogie, ceux de l'informatique et enfin les personnes handicapées mentales.

### 3.3. La démarche suivie

#### 3.3.1. La démarche

Elle ne constitue pas une surprise et se base donc sur les quatre idées de base évoquées à la section 3.1.1. et sur les principes de la section 3.2. sur la participation des utilisateurs.

Dans un premier temps, nous jouons un rôle passif, dans la relation avec les psychologues. Ceux-ci expriment les capacités, les caractéristiques de la population, ainsi que ce qu'ils voudraient que l'application interactive réalise (bien entendu à ce stade la définition des besoins est encore vague, et devra être affinée par itérations).

Parallèlement nous jouons aussi un rôle passif dans la relation avec les personnes handicapées mentales, en participant à leurs activités.

Ensuite la collaboration devient plus équilibrée, sur les deux plans ; avec les psychologues nous spécifions les fonctionnalités envisagées a-priori, de même que les messages interactifs utiles à la réalisation de la tâche. Puis nous imaginons les présentations possibles de ces messages par des objets interactifs et par conséquent les actions qui seront demandées aux futurs utilisateurs (par exemple si un message sera représenté par une icône, cela implique que l'utilisateur devra cliquer avec la souris sur cet objet pour réaliser une opération associée au message interactif représenté par l'icône).

Ensuite avec les personnes handicapées nous faisons des tests, des expérimentations sur des petits prototypes, pour voir si les présentations imaginées sont compréhensibles et si les actions prévues, les fonctionnalités attendues en somme, sont réalistes et pourront être effectuées par les utilisateurs.

On comprend bien qu'après ces tests on revient à la phase de spécification avec les psychologues (c'est le processus itératif) pour modifier les spécifications de l'interface, des fonctionnalités ou des deux.

Il faut ajouter que parallèlement à ces étapes, il est nécessaire de connaître les possibilités offertes par le matériel utilisé, puisque finalement il faudra réaliser un programme, et qu'il est tributaire de ce que le hardware et le software choisis permettent. Ils établissent la limite de la

faisabilité informatique.

### 3.3.2. Commentaires sur la démarche

Tout d'abord nous tenons à préciser que cette démarche ne prétend pas être une méthodologie rigoureuse, à suivre pour la phase de spécification d'une application interactive destinée à des personnes handicapées mentales. Nous essayons simplement de relater de manière un peu formelle l'expérience que nous avons vécue lors de cette première phase du cycle de vie du projet. Ce que nous voulons montrer c'est que cette phase ne doit pas être le fruit du hasard, et que cette démarche itérative, qui peut sembler inutilement lente (notamment toute la partie où notre rôle est délibérément passif), nous paraît très importante.

Ensuite, nous voudrions faire remarquer que dans cette phase, les spécifications des fonctionnalités et des interfaces possibles sont menées en parallèle (comme souhaité dans la section 3.1.3.), et que par le processus itératif, elles s'affinent au fur et à mesure des tests menés.

A propos de tests, il faut préciser que nous sommes conscients de les avoir mené auprès d'un petit "échantillon" de personnes handicapées, et que par conséquent il faut être prudent si on veut généraliser les résultats. Toutefois s'il est vrai que les expérimentations ont été faites auprès d'un petit nombre de personnes, ces personnes donnent une assez bonne représentation des divers niveaux de handicap mental. Ensuite il faut dire que nous avons aussi les rapports concernant les activités faites dans les institutions belges avec des ordinateurs (dont certains résultats ont été confirmés). Et enfin, il faut savoir que c'était la seule façon que nous avions de travailler dans le temps qui nous était imparti. De plus, les psychologues étaient aussi garants de la validité des résultats.

Notre quatrième remarque se rapporte au fait que la démarche suivie permet bien de prendre en compte les quatre idées de base qui ressortent des travaux sur la spécification d'applications interactives, à savoir une approche itérative, une analyse des fonctionnalités et des interfaces, se basant sur la connaissance des utilisateurs.

### 3.4. Comment s'est passé concrètement cette phase

Dans le chapitre suivant nous verrons en détail les spécifications auxquelles nous sommes arrivés, tant au niveau des fonctionnalités qu'au niveau des interfaces, le but de cette section est de voir comment la démarche présentée fut mise en application, de voir concrètement les "itérations" expliquées ci-dessus. C'est lors de ce processus que se sont effectués des choix (et par conséquent des rejets) de tel ou tel type de dialogue, nous verrons donc ce qui a motivé ces choix.

Sans entrer dans le détail, disons que le programme auquel nous sommes arrivés veut apprendre aux personnes handicapées mentales à tenir leurs comptes (comptabiliser leurs dépenses et leurs recettes), ce qui est le premier pas avant toute activité de prévision budgétaire (par exemple dans les entreprises, toute comptabilité budgétaire n'est possible que sur base d'une bonne comptabilité, donc des enregistrements des dépenses et des recettes).

#### 3.4.1. En Belgique

Comme nous l'avons dit dans le chapitre 2, ce travail s'est basé sur une première réalisation faite dans le cadre du PSINHA, au Département de Psychologie des Facultés. Par conséquent, la phase de spécification n'est pas partie de rien. Il fallait profiter du travail réalisé par l'équipe du PSINHA. De plus, si cette phase devait se passer principalement dans le cadre de notre premier stage dans l'institution La Castalie, en Suisse, il est clair qu'elle commençait en Belgique.

Donc notre première tâche fut de reprendre tous les rapports du groupe de travail qui avait participé à l'élaboration du premier programme, ainsi que les spécifications auxquelles il était arrivé.

Bien entendu notre travail ne consisterait pas à réécrire le programme à partir des mêmes spécifications, sur une autre machine avec de meilleures possibilités graphiques et sonores. Il s'agissait de se baser sur ce travail, d'en reprendre les forces et d'essayer d'en améliorer les faiblesses. Ces forces et ces faiblesses étant à relever tant au niveau des



fonctionnalités qu'au niveau de l'interface. C'est donc pour cela qu'outre les spécifications finales, nous avons recherché le maximum de documents, de rapports qui disaient (explicitement ou non) les choses qui semblaient cruciales et celles qui étaient criticables.

Toutefois, il faut préciser qu'avant de partir pour la Suisse, nous n'avions pas encore rencontré de personnes handicapées mentales ; une lacune que n'aurait pu combler aucun rapport écrit.

### 3.4.2. En Suisse

#### 3.4.2.1. Passifs

Voyons maintenant comment dans un premier temps nous avons joué un rôle passif dans notre collaboration avec les psychologues et les futurs utilisateurs.

Nous sommes donc allés un mois à La Castalie qui a été présentée au chapitre 2. Comme prévu, lors des premières réunions avec les psychologues, ceux-ci étaient beaucoup plus actifs que nous.

Tout d'abord nous avons pris connaissance des expériences qui avaient été menées avec des ordinateurs ou des machines à calculer dans diverses institutions avec des adolescents ou des adultes. Donc, chaque psychologue, ayant vécu ces expériences, a pu mettre en évidence les avantages, les inconvénients et les possibilités de telles méthodes. C'est de ces discussions, entre autres, que nous avons pu tirer les avantages de neutralité de l'ordinateur (moins "sévère" que l'éducateur), les aspects affectifs qu'il permet de renforcer, l'attrait qu'il suscite auprès des personnes handicapées, que nous avons présentés à la section 1.1.3. sur l'évaluation de l'emploi de l'informatique dans le domaine du handicap mental.

Ensuite les psychologues ont fait un premier relevé des choses qu'ils souhaiteraient voir réalisées par un programme d'aide à la gestion d'argent. Bien entendu, il était clair que d'une part, toutes les idées ne pourraient être implémentées dans le temps qui nous était imparti (mais ceci n'était pas important et n'empêchait pas de recueillir toutes les propositions, car à ce moment là il n'était absolument pas question de choisir ce qui devait être réalisé en premier), et d'autre part, ces besoins

étaient énoncés de façon vague et informelle, par conséquent les spécifications découleraient d'un processus itératif d'affinements successifs.

Nous voudrions insister sur le fait qu'aucun choix ne doit être fait à cette étape, et que toutes les idées fondées doivent être retenues, puis c'est au fil des "itérations" qu'on peut déceler celles qui sont éventuellement à rejeter (pour cause d'impossibilité avec le matériel à disposition, ou pour cause d'incompatibilité avec la population visée).

Outre ce premier jet (en fait deuxième, puisque le premier programme servait de base à ce travail) sur les fonctionnalités, il y eut un premier relevé des techniques possibles de dialogue. Par exemple nous avons répertorié les représentations possibles de la notion de "dix francs" :

- \* le nombre en chiffres : "10"
- \* le nombre en lettres : "DIX"
- \* le billet de dix francs (pour la Suisse) ou ses équivalents en pièces
- \* les mots "dix francs" dits (par exemple par une synthèse vocale)
- \* et aussi graphiquement, par une petite bande de couleur qui aurait par exemple la moitié de la longueur d'une bande représentant 20 francs (nous reviendrons sur cette représentation graphique, donnant une idée des proportions relatives de telle ou telle somme par rapport à une somme totale. Nous avons vu au chapitre 2 que c'est une des idées importantes du premier programme).

Donc pour résumer, au cours de ces réunions, les premières définitions des besoins, des modes de dialogue possibles étaient exprimés. De même que les avantages de l'utilisation de l'ordinateur, et les caractéristiques générales de la population handicapée mentale. Ce fut réellement une phase d'apprentissage pour nous.

Parallèlement, nous prenions nos premiers contacts avec les handicapés mentaux. Nous avons suivis deux groupes dans des activités diverses. Tout d'abord aller faire des achats dans un centre commercial ; l'éducatrice disait simplement à chacun les articles qu'il devait se procurer, puis ils partaient à la recherche dans les rayons. Sachant qu'aucune personne du groupe ne savait lire, la rapidité avec laquelle ils se sont tous acquités de leur tâche était réellement surprenante.

Nous avons également suivi des activités à caractère plus éducatif,

telles que des exercices d'écriture (recopier à la main un texte modèle) et de lecture pour les meilleurs. Un exercice a particulièrement retenu notre attention : une personne, qui ne savait pas compter, devait faire le total de l'argent dont elle disposait (uniquement des pièces suisses de 5 centimes à 5 francs). Elle avait une calculatrice devant elle, mais aussi des cartons représentant les diverses touches de la machine à calculer. En premier lieu, elle devait placer sur la table les cartons représentant la séquence des touches sur lesquelles elles devait appuyer pour additionner une pièce au total actuel. Puis, si c'était correct, elle effectuait réellement la tâche.

Ce système des cartons était bien sûr destiné à diminuer le nombre d'erreurs qu'elle aurait sans doute commises en utilisant directement la machine. Toujours est-il que malgré quelques erreurs, la personne a fini par trouver le total exact.

Donc cette observation nous a montré que même en ne sachant pas compter, des personnes handicapées mentales peuvent manipuler une machine qui le fait à leur place et enregistrer de la sorte tout leur argent (or c'est bien une des choses envisagées par le programme).

Nous avons assisté à d'autres exercices sur l'argent, par exemple sur le problème de l'équivalence de l'argent, qui est très complexe (et d'autant plus en Suisse où la monnaie est telle que les centimes sont fréquemment utilisés). Savoir qu'une pièce de 1 franc est équivalente à 2 pièces de 50 centimes n'est pas une chose aisée.

D'autres expériences intéressantes mêlaient vie quotidienne, manipulation d'argent et de machine à calculer. Un groupe de personnes se rendait dans un centre commercial et allait chercher les divers articles dans les rayons. Ensuite, avant de passer aux caisses, à l'aide de machines à calculer, ces personnes faisaient le total, préoyaient le billet avec lequel payer et calculaient le montant qu'elles devaient recevoir en retour (ce genre d'expérience est également mené en Belgique, dans les institutions qui ont collaboré au premier travail).

Et finalement, nous avons assisté à des travaux sur ordinateur (donc nous avons la chance de travailler avec des personnes qui n'étaient pas vierges du point de vue informatique). Il s'agissait de dactylographier des étiquettes sur base de modèles. Ceci prouvait que des personnes ne sachant

pas lire pouvaient utiliser un clavier et pouvaient recopier un texte modèle. Plus surprenant encore, toutes étaient capables de taper leur prénom, sans modèle. Ceci nous a permis de poser des hypothèses au niveau des fonctionnalités du programme, comme nous le verrons dans la section suivante.

En conclusion, ces expériences nous ont permis de prendre contact avec la population handicapée, de voir déjà ce qu'il était possible de faire faire même aux moins doués, dans le domaine de l'argent, et avec un clavier (de machine à calculer ou d'ordinateur). De plus, nous avons pu voir l'attrait que suscitait une simple machine à calculer, ce qui est encourageant puisqu'a-priori, un ordinateur avec un écran plus grand, des couleurs et du son devrait être encore plus attrayant.

C'était donc la phase où nous étions le plus passif et où nous suivions un véritable apprentissage, absolument nécessaire.

#### **3.4.2.2. Actifs**

Ensuite notre rôle est devenu plus actif tant dans notre collaboration avec les psychologues qu'avec les futurs utilisateurs. Avec les psychologues, nous avons donc définis les fonctionnalités qui paraissaient possibles au vu de la population visée et imaginé des formes de dialogue. Sans entrer dans les détails des spécifications, qui seront présentées au chapitre suivant, comme fonctionnalités, nous avons prévu un logiciel qui permette aux personnes handicapées de tenir leurs comptes (leur comptabilité en somme), et qui leur apprenne donc à gérer l'argent dont ils disposent pendant un certain laps de temps, par exemple une semaine (ce que nous avons appelé un cycle). Donc pouvoir enregistrer leurs dépenses et leurs recettes, tout en cherchant à développer les notions de temps (justifier le moment où ils ont reçu ou dépensé de l'argent, mais aussi se rendre compte de l'argent qu'il leur reste par rapport au temps qu'il reste avant la fin du cycle), de justification des dépenses. Dans les spécifications, nous avons donc prévu qu'enregistrer une dépense consiste non seulement à donner le montant qui doit être comptabilisé, mais aussi le moment auquel a été faite la dépense, ainsi que le "poste budgétaire" correspondant, c'est-à-dire le bien ou le service acheté.

Nous en sommes arrivés à la même idée pour les recettes, mais bizarrement dans un premier temps nous avons considéré qu'il suffisait d'introduire le montant (sans donner le moment et la provenance de la recette). C'est un exemple de l'affinement des spécifications qui fait qu'il est presque impossible d'obtenir les spécifications définitives au premier jet.

De plus, si le but est d'amener les utilisateurs à tenir les comptes de leurs dépenses et recettes à jour, il faut que le programme dise à l'utilisateur si ses comptes sont à jour, c'est-à-dire s'ils sont le reflet de la réalité, de ce qu'il possède réellement. Par conséquent, pour pouvoir faire cette comparaison entre la situation comptable et la réalité, l'ordinateur doit connaître la somme d'argent que possède réellement l'utilisateur quand il vient faire une session de travail pour tenir ses comptes. Donc, dès le début d'une séance de travail, l'utilisateur doit communiquer à la machine la somme d'argent dont il dispose réellement. Cette fonctionnalité n'était pas apparue d'entrée de jeu, et elle émane donc de la réflexion menée avec les psychologues. Mais remarquons qu'elle est essentielle, car sans cela il est impossible que le programme sache si l'utilisateur a mis ses comptes à jour ou non. Toutefois, si la somme communiquée au début est fausse, il est clair que tout ce que conclura le programme, quant à la tenue des comptes sera faux, c'est une des limites de tout programme de gestion d'argent, nous en reparlerons au chapitre suivant.

Une autre phase importante des fonctionnalités, est celle initiale où l'utilisateur s'identifie, elle est nécessaire pour que le programme sache quel "livre de comptes" il faut manipuler. Suite à nos observations, nous savions que nous pouvions demander aux utilisateurs de commencer par taper leur prénom. On perçoit de nouveau l'approche itérative, puisque les observations (qui sont en somme déjà des tests) faites lors de la phase passive permettent déjà de faire des hypothèses sensées au niveau de la spécification des fonctionnalités.

Mais si nous avons parlé des fonctionnalités, nous n'avons pas encore beaucoup abordé le problème des interfaces. En fait nous avons vu au chapitre précédent que la machine utilisée offrait des possibilités au niveau sonore et graphique, et qu'outre le clavier, il serait possible d'utiliser la souris dans une optique de manipulation directe (voir section

#### 1.2.2.1. sur les techniques d'interaction).

Par conséquent, l'idée était de mettre en oeuvre ces possibilités de façon à offrir plusieurs dialogues pour le même jeu de fonctionnalités. Donc avec les psychologues nous avons pensé qu'il fallait proposer plusieurs éléments de dialogue qu'il serait possible de combiner au mieux selon les caractéristiques de la personne qui va utiliser le programme. Donc très vite est venue l'idée que le programme pourrait être paramétré, c'est-à-dire que l'éducateur choisit pour chaque personne les paramètres qui permettront d'avoir les éléments de dialogue les mieux adaptés.

Qu'entendons-nous par "éléments de dialogue" ?

En fait il s'agit d'objets interactifs, qui peuvent être de saisie ou d'affichage (cela comprend aussi le son comme nous l'avons déjà fait remarquer), et qui sont la "traduction" des messages interactifs. Nous avons vu que dans la méthodologie de spécification proposée par J.C. Ruche, il y avait une distinction entre la tâche, qui est la définition abstraite de l'interface, en termes de messages interactifs, et qui ne fait aucune hypothèse sur les représentations de ces messages, et la présentation concrète via des objets interactifs utilisant les diverses techniques d'interaction. En pratique, il nous a été difficile de maintenir une telle dichotomie, comme nous l'avons vu à la section 3.1.2.3.

Concrètement, une fonctionnalité de l'application est d'enregistrer le montant d'une recette ; il a été pressenti que l'utilisateur pourrait le faire en introduisant un nombre au clavier ou bien en désignant un à un les billets et pièces composant la somme d'argent, et cela en cliquant avec la souris sur des représentations de ces billets et pièces affichés à l'écran. Donc on peut dire que le message interactif "montant-recette" serait dans un dialogue représenté par un formulaire (limité à un seul champ, qui serait donc complété via le clavier), et dans l'autre cas il serait "introduit" en plusieurs étapes par manipulation directe d'icônes.

Toutefois ce n'est pas tout à fait exact, parce qu'à ce moment-là, nous n'avions pas fait de supposition quant à la présentation de ces icônes. Donc le dialogue par "billets et pièces" est quand même une sorte de dialogue abstrait (même si à un degré d'abstraction moindre). Nous avons pu raisonner et construire ce que nous avons appelé les "scénarios abstraits de chaque fonction", et puis seulement nous nous sommes posé la question des

présentations concrètes à l'écran. Ceci est peut-être d'une importance toute relative mais nous voulions y revenir un instant.

Donc nous avons imaginé diverses présentations des messages interactifs. Voyons quelques exemples.

Poser une question à l'utilisateur telle que "combien avez-vous dépensé ?" peut prendre plusieurs formes ; la phrase peut être écrite à l'écran (puisque certaines personnes handicapées savent lire), elle peut être prononcée par la synthèse vocale, ou évoquée par une icône.

Pour les saisies de sommes, comme nous l'avons dit, elles peuvent se faire par le clavier ou par la souris en cliquant sur des représentations de billets et pièces (nous verrons les trois présentations de ces icônes que nous avons imaginées).

Au niveau des feed-back, qui sont très importants, nous en avons imaginé trois types ; les feed-back acoustiques, c'est-à-dire qu'une phrase est prononcée par la synthèse vocale suite à une action, les feed-back graphiques (des icônes apparaissant suite à une action, des bandes de couleurs augmentant ou diminuant proportionnellement à une valeur de référence...), et enfin des feed-back textuels ou numériques, par exemple un total affiché suite à une addition.

#### **3.4.2.3. Tests**

Bien entendu tout ceci relevait d'un travail avec les psychologues, mais devait nécessairement passer par l'expérimentation pratique. Donc nous avons réalisé des petits prototypes, des petits tests destinés à voir si des fonctionnalités et des éléments de dialogues prévus, étaient réalistes pour les futurs utilisateurs. Nous avons menés ces tests auprès d'une dizaine de personnes (nous avons déjà parlé du problème de la généralisation des résultats). Il ne s'agissait pas de programmes, mais d'écrans statiques qui permettaient d'être vite fixé sur le réalisme de telle ou telle hypothèse. Voyons quelques exemples d'expérimentations que nous avons effectuées.

Lors de nos tests nous avons demandé aux jeunes de taper leur prénom au clavier, ils le firent tous, ce qui confirmait que nous pouvions commencer le programme par ce genre d'identification. Comme ils étaient

aussi capables de recopier des modèles (pour les étiquettes), cela montrait qu'ils pouvaient, par exemple, enregistrer le montant d'une dépense qui se trouve sur un ticket de caisse en le recopiant au clavier. Donc cela confirmait la possibilité d'un tel dialogue.

Nous avons aussi réalisé des digitalisations "sonores" de quelques courtes phrases, qui nous ont montré que la synthèse vocale était utilisable même si la compréhension n'était pas toujours immédiate. En fait, ceci nous a amené un peu plus tard à ajouter sur certains écrans une icône permettant de réécouter le dernier message prononcé (encore le résultat du processus itératif).

Au point de vue des présentations des billets et pièces, nous avons imaginé trois possibilités : des digitalisations (que nous avons réalisées avec une caméra et l'Amiga), des dessins ressemblants avec des couleurs, et enfin des billets et des pièces dessinés de façon tout à fait symbolique, c'est-à-dire des rectangles et des cercles contenant uniquement des nombres. Nous avons alors donné de vrais billets et pièces à chaque personne qui venait (il est à noter que même celles qui n'avaient jamais vu de billets savaient qu'il s'agissait d'argent). Puis nous avons présenté différents écrans où se trouvaient tantôt des digitalisations, tantôt des dessins, tantôt les deux, et nous leur avons demandé de nous montrer si quelque chose sur l'écran ressemblait au billet ou à la pièce qu'elles avaient en main.

Les résultats furent excellents et un peu surprenants. Toutes les personnes ont reconnu tous les billets et pièces et ce dans tous les modes de présentation proposés. C'est très important parce qu'il s'avère que des digitalisations complexes et coûteuses ne sont pas obligatoires puisqu'elles ne donnent apparemment pas de meilleurs résultats que les dessins les plus symboliques.

Bien entendu, nous ne prétendons pas que les résultats des tests sont indiscutables, toutefois de l'avis des psychologues avec qui nous étions, et vu la diversité de handicap que présentaient les personnes qui ont participé aux tests (de façon individuelle), les résultats sont acceptables. Nous avons donc décidé, dans un premier temps de proposer ce type de dialogue avec des billets et des pièces sous forme symbolique.

Une autre expérimentation portait sur la manipulation de l'ordinateur,



car si nous avions la preuve que le clavier était "maîtrisé", la souris par contre n'avait encore jamais été approchée. Nous avons demandé aux jeunes de déplacer le suiveur de la souris à tel ou tel endroit de l'écran, après leur avoir fait une démonstration. Les résultats furent absolument étonnants. Nous avions prévu une grande difficulté due aux deux plans perpendiculaires de l'écran et de la souris, or dans la plupart des cas l'apprentissage fut immédiat. Dans quelques cas se posait un problème de lenteur de manipulation, mais jamais d'orientation, ce qui est tout à fait surprenant quand on voit la difficulté avec laquelle beaucoup de personnes non handicapées apprennent à manipuler cet instrument. Donc, même si certaines personnes agissaient lentement, toutes pouvaient amener le suiveur de la souris à n'importe quel endroit de l'écran. Cependant, le suiveur de la souris de l'Amiga est représenté par une flèche oblique et nous avons perçu que certains handicapés éprouvaient des difficultés à comprendre que c'est la pointe du suiveur de la souris qu'il fallait placer sur l'objet pour le sélectionner. Ces personnes plaçaient le centre de la flèche sur l'objet et si celui-ci était de petite taille, la sélection échouait. Nous avons compris qu'il était préférable d'avoir un suiveur dont le centre était le point de sélection. Les tests suivants que nous avons menés proposaient un suiveur en forme de gros "+" dont le croisement des deux branches était le point de sélection. Il n'y a plus eu de problème de ce type à partir de ce moment.

Par la suite nous avons refait l'exercice des billets et pièces à reconnaître (sous n'importe quelle forme), et nous demandions de désigner la ou les bonnes représentations non plus du doigt, mais en y amenant le suiveur de la souris. Ce test fonctionna très bien, et cela démontrait qu'il serait possible à des utilisateurs de désigner par exemple tous les billets qu'ils ont dans leur porte-monnaie en cliquant sur leurs représentations, une à une (et cela n'implique absolument pas de savoir compter).

Pour être encore plus convaincu, nous avons refait des tests en proposant un dessin d'écran interactif (qui apparaîtrait dans l'application) que nous avons imaginé. Il contenait uniquement des dessins symboliques de pièces et de billets puisque suite à nos premiers tests, ce mode de présentation s'était avéré efficace. Et nous avons fait des exercices

consistant à désigner avec la souris un tas de pièces et de billets que nous mettions sur la table. Bien sûr il n'y avait aucun "effet", aucun feed-back, puisqu'il s'agissait d'un dessin statique, d'un prototype rapide d'un des écrans que nous imaginions.

Voilà donc quelques exemples de tests qui ont été faits avec les utilisateurs, et qui ont servi à définir de plus en plus finement les fonctionnalités et les modes de dialogue qu'il serait réaliste de mettre en oeuvre.

Cette phase de double collaboration avec les psychologues et avec les utilisateurs nous paraît absolument nécessaire (mais il faut rester conscient que cela ne garantit pas la réussite du projet à coup sûr, et que les spécifications risquent encore de "glisser" au fur et à mesure que le projet avance).

## Chapitre 4. Les spécifications du programme

### Introduction

Tandis que le premier chapitre présentait le domaine de l'informatique et du handicap et celui de l'interface homme-machine, le chapitre 2 situait le contexte de ce travail et le chapitre 3 donnait les fondements de notre démarche de spécification.

Le but de ce chapitre est tout d'abord d'apporter quelques précisions sur les buts et les limites du programme, ensuite de donner les spécifications (des fonctionnalités et de l'interface) auxquelles nous sommes arrivés. Nous verrons aussi celles du programme auxiliaire servant à choisir les éléments de dialogue, et enfin nous verrons en quoi les règles et les principes d'interface évoqués au chapitre 1 sont respectés ou non.

### 4.1. Les buts et les limites du programme

Dans cette section, nous allons essayer de donner quelques précisions sur le projet auquel nous avons contribué ; tout d'abord du point de vue pédagogique et ensuite du point de vue des limites intrinsèques au programme.

#### 4.1.1. D'un point de vue pédagogique

##### 4.1.1.1. Le projet répond à quels besoins ?

Comme nous l'avons vu au chapitre 2, le premier travail répondait aux demandes exprimées par deux institutions belges ayant fait appel au PSINHA. Le but était d'aider des personnes handicapées mentales adultes, relativement autonomes, à gérer leur budget hebdomadaire ou mensuel. Ensuite, nous avons essayé d'étendre la portée du programme, afin de répondre à un plus grand nombre d'institutions potentiellement demandeuses ; c'est-à-dire qui ont la volonté de faire comprendre les notions de budget à gérer pendant un laps de temps, de recettes, de dépenses, de tenue des comptes, de justification des comptes ....etc.

#### 4.1.1.2. Quels sont les objectifs ?

Il est difficile et dangereux de vouloir définir de façon définitive les objectifs poursuivis, toutefois on peut en exprimer deux principaux :

\* **premièrement, il s'agit de fournir un outil d'apprentissage; en effet le programme devrait permettre à une personne handicapée mentale de saisir progressivement les notions de budget, de recettes, de dépenses, de tenue des comptes ....etc.**

\* **deuxièmement, il s'agit de fournir un outil d'autonomie ; c'est-à-dire permettant aux handicapés (ou du moins à certains d'entre eux) de gérer réellement leur budget de manière autonome ou presque.**

De plus, il est certain que l'utilisation du programme peut avoir des retombées positives au niveau affectif, ou au niveau de la manipulation de l'ordinateur, mais il ne s'agit pas d'objectifs premiers de ce projet.

Mais pour conférer un caractère ouvert au logiciel, nous ne devons en aucun cas fixer un ordre des étapes qui doivent être parcourues (par exemple, pour passer d'une phase d'apprentissage à une phase d'autonomie). En effet, certaines personnes resteront peut-être toujours au stade de l'apprentissage, ou pourront travailler de façon autonome pour certaines notions bien comprises et pas pour d'autres.

Pour illustrer ce point, voici un exemple que nous avons vécu :

Aux EPSE La Combe à Genève, nous avons rencontré un groupe d'adultes (qui travaillaient dans un atelier de menuiserie). Toutes les personnes avaient la notion d'économie, c'est-à-dire qu'il faut mettre de l'argent "de côté" pour faire un achat plus conséquent, mais certaines personnes étaient incapables de reconnaître les pièces de monnaie. Or, quelques jours plus tôt, lors de discussions avec les psychologues sur la possibilité de fixer un ordre des phases d'apprentissage, nous pensions qu'au niveau le plus bas, il fallait proposer des fonctionnalités demandant uniquement de savoir reconnaître l'argent, puis au niveau suivant on intégrerait des fonctions qui supposeraient de savoir ce que l'on peut acheter avec l'argent dont on dispose, et au niveau le plus élevé, il serait possible de prendre en compte les fonctions de prévisions de dépenses et de recettes en vue d'un achat plus conséquent.

Donc, nous nous sommes largement trompés, puisqu'il est possible de

posséder des notions que nous croyions les plus difficiles à acquérir et pas celles que nous pensions être les plus simples. Ceci montre donc qu'il est inopportun de vouloir fixer un ordre des concepts qui doivent être acquis pour passer d'une phase d'apprentissage à une phase d'autonomie.

Ce que nous souhaitons, c'est proposer un certain nombre d'éléments de dialogue, que l'éducateur combinera en fonction des aptitudes particulières de la personne destinée à utiliser le programme (mais sans supposer que tel élément ne peut être employé que si tel autre est maîtrisé). Ces choix seront faits grâce au programme de paramétrisation, dont les spécifications seront présentées à la section 4.3.

#### **4.1.1.3. A quelle population le projet est-il destiné ?**

Il est destiné à des personnes handicapées mentales à partir d'une quinzaine d'années, qui disposent d'une certaine somme d'argent (soit reçu : argent de poche, soit gagné : rémunération d'un travail presté).

Comme nous l'avons annoncé dans le chapitre 1, nous ne voulons pas classer les utilisateurs selon une échelle de quotient intellectuel, qui nous paraît inadaptée au problème.

On veut pouvoir prendre en compte aussi de légers handicaps physiques (troubles légers de la vision et de la motricité), notamment en utilisant des caractères d'affichage suffisamment grands.

Remarquons qu'une amélioration pour des personnes sourdes connaissant un langage signé, est envisagée. Nous en reparlerons au chapitre 7.

En ce qui concerne les handicaps moteurs plus profonds, ces problèmes ne peuvent être réglés que par des moyens auxiliaires tels que ceux développés par la FST.

#### **4.1.1.4. Quels sont les prérequis ?**

Nous allons énumérer un certain nombre de choses qui nous paraissent importantes. Toutefois, il faut rester prudent car nous avons vu à la section 4.1.1.2. que des hypothèses apparemment sensées peuvent être tout à fait infirmées dans la réalité. Donc, il est possible que des personnes, ne réunissant pas tous les prérequis suivants, puissent tout de même

profiter de l'utilisation du programme. Voici donc une liste de prérequis :

- \* sensoriels : reconnaissance des formes et des objets.
- \* moteurs : possibilité d'utiliser un clavier et/ou une souris, puisque, dans un premier temps, ce seront les outils que le programme mettra en oeuvre (mais comme nous l'avons vu, il n'est pas exclus de penser à utiliser un crayon graphique ou un écran tactile).
- \* intellectuels : la compréhension du langage parlé, dans une bonne mesure, paraît indispensable. Nous avons ensuite émis l'idée d'une possibilité limitée de comptage (jusque 3 ou 4) mais nous préférons ne pas insister sur un tel prérequis.
- \* culturels : savoir qu'on n'a rien pour rien et qu'il faut de l'argent pour se procurer quelque chose. Savoir plus ou moins facilement reconnaître les billets et les pièces du pays.
- \* affectifs : nous postulons que les personnes qui seront concernées seront dans de bonnes dispositions. Cependant, nous sommes conscients qu'il peut y avoir des blocages affectifs (par exemple le fils de banquier qui voulait amasser de l'argent et n'en dépensait jamais...). Nous supposons que ces blocages sont "dépistés" et vaincus par ailleurs. Le projet ne les intègre pas. Toutefois il s'est déjà avéré que de tels blocages pouvaient être vaincus par l'utilisation de l'ordinateur, qui fait disparaître les relations interpersonnelles. Si de tels cas se manifestaient, nous ne pourrions que nous en réjouir.

#### 4.1.1.5. Dans quel cadre sera-t-il utilisé ?

Principalement dans le cadre des institutions d'accueil. De plus, l'achat de l'ordinateur est plus facilement consenti par une organisation que par une famille. Toutefois, la machine choisie reste dans une fourchette de prix abordable, qui n'exclut pas qu'une famille dispose d'un ordinateur et utilise le programme.

#### 4.1.2. Limites de validité de tout programme d'aide à la gestion d'argent

Cette section ne comporte que des choses évidentes, mais nous pensons qu'il est intéressant de les mettre noir sur blanc une fois pour toutes.

Une grosse différence entre l'ordinateur et l'éducateur, en ce qui concerne la gestion du budget d'une personne handicapée, réside dans le fait qu'à tout moment, ce dernier peut compter ce que la personne possède et vérifier si les sommes d'argent qu'elle communique à l'ordinateur sont correctes. Il peut sembler stupide de s'attarder sur ce point, mais en fait il est nécessaire de comprendre que l'ordinateur ne pourra jamais le faire : en effet, on peut imaginer n'importe quelle astuce pour saisir une somme d'argent, si par exemple la personne oublie un billet (plus ou moins volontairement), la machine ne le saura jamais et toutes les opérations qu'elle ferait sur ces nombres, aussi précises soient elles, seraient tout à fait fausses et sans rapport avec la réalité. C'est une limite et il faut en être conscient. Cependant, nous avons vu lors de la spécification des fonctionnalités (section 3.4.2.2.) qu'une certaine vérification de l'exactitude des opérations effectuées par l'utilisateur est possible. Il faut pour cela, qu'en début de session, la personne "introduise" la somme dont elle dispose, de sorte que l'ordinateur puisse vérifier l'écart qui existe à tout moment entre le solde des comptes et la réalité ; c'est-à-dire voir si les comptes sont le reflet de la réalité (quand tout est bien comptabilisé, on a l'égalité suivante :

Solde des comptes de la session précédente

+ Somme des recettes enregistrées pendant cette session

- Somme des dépenses enregistrées pendant cette session

= Somme introduite (dont l'utilisateur dispose) en début de session).

Bien entendu, il est clair que si la somme entrée est fausse, l'ordinateur croira toujours détecter un oubli alors qu'en fait toutes les dépenses et toutes les recettes auront été introduites correctement. Mais si la première somme est exacte, un certain nombre d'oublis pourront être repérés.

Nous nous contenterons donc de demander à l'utilisateur d'introduire la somme dont il dispose en début de session. A titre indicatif, voici deux

systèmes qui ont été imaginés lors de discussions avec les psychologues. On peut imaginer un système "avaleur" où l'on introduit pièces et billets, et qui les rend après les avoir identifiés, ou encore un système de caméra qui identifierait les pièces et les billets que l'utilisateur placerait sur une table visée par l'objectif. Bien entendu, même ces systèmes ne sont pas sûrs à 100%, car si une pièce n'est pas introduite dans l'avaleur ou n'est pas placée sur la table, elle sera ignorée et des erreurs en découleront.

## 4.2. Les spécifications du programme

Cette section a pour but d'exposer les spécifications que nous avons élaborées en suivant la démarche expliquée dans le chapitre 3. Nous exposerons d'abord les spécifications des fonctionnalités et les objets abstraits d'interface nécessaires au déroulement correct des fonctionnalités, ainsi que le scénario du dialogue qui a lieu pour mener à bien la tâche (ce qui correspond plus ou moins à la spécification des messages interactifs et au schéma de la conversation, que J.C. Ruche préconise dans la phase de spécification de la tâche). Ensuite, nous verrons les objets concrets correspondants que nous avons choisis (il s'agit donc des objets interactifs qui sont la présentation des messages interactifs définis lors de la spécification de la tâche).

Comme dit dans l'introduction de ce mémoire, il ne nous a pas été possible de développer la totalité des spécifications, faute de temps. Nous indiquerons donc également dans cette section quelle est la partie que nous avons spécifiée en détail et implémentée.

### 4.2.1. Les spécifications des fonctionnalités

Nous avons ainsi défini cinq grandes fonctions :

- \* La tenue des comptes qui est donc une gestion a-posteriori, c'est-à-dire après que les recettes et les dépenses aient effectivement eu lieu. Il s'agit en fait d'une comptabilité, d'une gestion de budget : l'utilisateur note ses recettes et ses dépenses (en les justifiant) et calcule quel est l'état de ses finances suite à ces opérations et cela dans une perspective temporelle : en effet il gère un budget donné sur un cycle donné (par exemple une semaine), ce qui signifie qu'il ne faut pas avoir épuisé



toutes ses ressources avant la fin du cycle .

- \* Le catalogue ; qui est une fonction ayant pour but de montrer à l'utilisateur ce qu'il peut acheter avec la somme totale dont il dispose au moment de l'activation de cette fonction.
- \* Le magasin ; qui doit permettre à l'utilisateur de simuler l'achat de différentes marchandises pour un total maximum égal à la somme dont il dispose actuellement. L'utilisateur devra pouvoir sélectionner des objets (c'est-à-dire les mettre dans son caddie), les dé-sélectionner (c'est-à-dire les remettre dans le rayon) et voir l'effet de ces opérations sur son budget.
- \* L'épargne ; qui constitue la gestion de la "tirelire" de l'utilisateur. Il faut que celui-ci puisse placer de l'argent dans sa tirelire, en retirer et voir combien d'argent elle contient.
- \* Le budget qui est une gestion de l'avoir a-priori c'est-à-dire dans un aspect prévisionnel. Il s'agit donc pour l'utilisateur de déterminer en début du cycle (un jour, une semaine ou un mois) les dépenses et les recettes qui sont certaines afin de savoir ce dont il peut disposer librement et pour ne pas tomber à court d'argent avant la fin du cycle.

Outre ces cinq fonctions, il faut :

- \* Identifier l'utilisateur afin de retrouver les fichiers qui lui appartiennent, principalement le fichier contenant le solde de ses comptes et celui contenant ses paramètres de dialogue. Nous avons choisi d'identifier l'utilisateur par son prénom. Il est clair que le prénom ne constitue en aucun cas un identifiant. Un numéro d'identification est selon nous inutile étant donné qu'il y a peu de risques que deux personnes au sein d'une même institution portent les mêmes nom et prénom. Cependant, il peut arriver que deux personnes aient le même prénom. Il doit donc être prévu que l'utilisateur puisse également taper son nom (non obligatoire) ou un ajout quelconque qui le distinguera d'une autre personne portant le même prénom.
- \* Proposer le choix à l'utilisateur entre les cinq fonctions spécifiées ci-dessus ainsi qu'une option lui permettant de quitter le programme.
- \* En fin de programme, il faut sauvegarder les données relatives à la session qui vient de se terminer, principalement :
  - le nouveau solde obtenu suite aux différentes recettes et dépenses

- la liste des manipulations de l'utilisateur tout au long de cette session, ce que nous avons appelé la trace des manipulations (voir à ce propos le chapitre 6).

Lorsque les spécifications de ces fonctions ont été écrites, nous avons remarqué qu'il nous serait impossible, faute de temps, de les implémenter toutes. Il a été décidé de développer un premier "sous-système utile", il s'agit de celui qui prend en charge les fonctions de "tenue des comptes", qui constituent en fait la base de fonctions plus complexes telles que l'épargne ou le budget. Comme nous l'avons déjà souligné, pour les entreprises, une comptabilité budgétaire ne peut être envisagée que s'il existe d'abord une comptabilité générale qui réalise une bonne "tenue des comptes". C'est ce qui a justifié le choix de ce premier sous-système utile. De plus, en ce qui concerne l'implémentation elle-même, la fonction tenue des comptes contient des sous-fonctions qui seront réutilisables telles que enregistrer une somme, saisir un jour de la semaine, un poste budgétaire, ...

Voyons tout de même ce à quoi nous sommes arrivés avec les psychologues pour toutes ces fonctions, même s'il est clair que seul le premier sous-système sera développé.

#### 4.2.1.1. La tenue des comptes

La logique de fonctionnement se calque en fait sur celle de la comptabilité dans un ménage. On pourrait résumer cela par l'exemple suivant :

solde au 23/10 .....	6400
<u>recettes</u>	
salaire.....24/10.....	+ 30 000
vente.....30/10.....	+ 7 500
<u>dépenses</u>	
achats ménagers.....25/10.....	- 8 000
tabac.....25/10.....	- 1 600
loyer.....29/10.....	-12 000
solde au 30/10.....	22 300

Ainsi, le solde au 23/10 représente le résultat de la session de comptabilité précédente et le solde du 30/10 est le résultat de la session de comptabilité que l'on vient de faire. Les différentes recettes et dépenses sont celles effectuées entre le 23/10 et le 30/10 et sont communiquées lors de la session du 30/10.

On peut donc faire un parallèle entre la tenue des comptes d'un ménage et celle que nous spécifions pour les handicapés mentaux. Néanmoins, on ne peut pas être sûr que l'utilisateur ne va pas oublier des recettes ou des dépenses. Dès lors, si le programme se déroule comme indiqué ci-dessus, on risquerait d'obtenir un solde faux, ce qui entraînerait des erreurs dans toutes les sessions ultérieures. Il faut donc éviter au maximum ce type d'erreur. Pour trouver la solution, on peut à nouveau observer ce qui se passe dans le cas de la comptabilité d'un ménage. La personne qui tient les comptes calcule le nouveau solde après avoir écrit toutes ses recettes et dépenses. Elle compare alors ce solde et ce qu'elle a réellement et, s'il existe une différence, elle cherche l'explication de celle-ci. Si, finalement, elle ne trouve pas, elle indiquera cette différence dans un poste appelé divers. On peut donc encore suivre cette logique dans le cas d'utilisateurs handicapés mentaux : il faut que le handicapé indique au départ combien il a dans son porte-monnaie afin de pouvoir déterminer à n'importe quel moment s'il existe une différence entre cette somme (que l'on appellera "somme initiale") et le solde actuel (solde ancien + total des recettes déjà enregistrées - total des dépenses déjà enregistrées). La communication de la somme initiale est absolument nécessaire car elle constitue le seul moyen de contrôle.

La fonction "tenue des comptes" devra donc offrir les fonctionnalités suivantes :

- Il faut tout d'abord enregistrer la somme que l'utilisateur a dans son porte-monnaie, c'est-à-dire la somme initiale.
- Une fois cette somme enregistrée, le programme devra offrir le choix entre plusieurs options ; la fin du déroulement associé à chaque option ramène l'utilisateur automatiquement à ce choix, sauf l'option FIN qui peut faire sortir du programme. Ces options sont :

- \* l'enregistrement d'une recette : il faut tout d'abord saisir l'origine de la recette. Parents, argent de poche, travail et économie sont les

quatre possibilités de poste de recette que nous avons retenues. L'étape suivante consiste à enregistrer quand cette recette a été faite ; ce peut être un jour de la semaine si le cycle budgétaire est d'une semaine ou un moment de la journée (matin, après-midi, soir) si le cycle est d'une journée (voir à ce propos le programme auxiliaire de paramétrisation, section 4.3.). Il faut enfin que l'utilisateur communique la somme qu'il a effectivement reçue. Il faut ajouter cette somme à l'état des comptes (l' "état des comptes" est un montant représentant à tout moment le solde de la session précédente + le total des recettes - le total des dépenses déjà communiquées).

- \* L'enregistrement d'une dépense : les étapes qui constituent la saisie d'une dépense sont les mêmes que dans le cas d'une recette. Les postes budgétaires retenus sont : la nourriture, les habits, le tabac, les loisirs, les transports, la tirelire, les magazines et les choses diverses. Evidemment, dans ce cas-ci, il faut retrancher la somme saisie de l'état des comptes.

remarque : il faut noter que l'on retrouve le poste "tirelire" dans les dépenses et dans les recettes. Etant donné que nous n'avons pas développé la fonction "épargne", nous avons ajouté ce poste qui permet une gestion très simplifiée de l'épargne : on peut mettre de l'argent de côté (dépenser pour la tirelire) et retirer une certaine somme de ses économies (faire une recette en provenance de la tirelire). Néanmoins, c'est toujours à la personne handicapée de gérer effectivement son épargne (à combien cette épargne s'élève-t-elle ? , combien peut-elle retirer ? , ...).

- \* Le rappel de la somme initiale (option que nous avons appelée "Etat du porte-monnaie"), c'est-à-dire la somme que l'utilisateur a communiquée en début de programme. On indiquera également si les comptes sont à jour, c'est-à-dire si l'état des comptes est égal à la somme initiale. Si ce n'est pas le cas, on indiquera la nature du déséquilibre : si l'état des comptes est supérieur à la somme initiale, des dépenses doivent encore être enregistrées; sinon, ce sont des recettes qui "manquent" et qui doivent donc être enregistrées.
- \* La sortie de la fonction "tenue des comptes". Si cette option est

choisie, on indiquera si les comptes sont à jour et, dans le cas contraire, la nature du déséquilibre. On laissera également le choix à l'utilisateur, ayant vu cet état de choses, de revenir dans la fonction ou de la quitter malgré un possible déséquilibre.

#### 4.2.1.2. Le catalogue

Cette fonction devra indiquer d'une façon ou d'une autre à l'utilisateur qu'avec la somme dont il dispose actuellement, il peut acheter X occurrences de tel type de produit dont le prix unitaire est Y. Ceci a pour but de montrer l'"équivalence produit" de l'argent, c'est-à-dire de concrétiser la notion d'argent en mettant en rapport une certaine somme d'argent et sa valeur marchande.

On aura donc une indication de la somme initiale suivie d'une liste de triplets [type de produit, nombre achetable, prix unitaire].

Bien entendu, il est nécessaire que le type et le prix des produits possibles soient adaptés à l'environnement de l'utilisateur handicapé. Ainsi, on ne proposera pas nécessairement des cigarettes à un non-fumeur, ni des produits que l'utilisateur n'achète jamais. De plus, il ne faut pas que le prix unitaire de tel produit proposé soit de 250 Frs, alors que le magasin auquel le handicapé fait habituellement ses courses vend le même produit 170 Frs (lors de nos activités avec les personnes handicapées, nous avons vu qu'elles peuvent attacher beaucoup d'importance à un prix, un chiffre qu'elles connaissent bien). De ce fait, il faudrait d'une part un utilitaire permettant à l'éducateur de spécifier de nouveaux produits, de changer le prix des produits existants et, d'autre part, une possibilité de choisir, parmi l'ensemble des produits existants, ceux qu'il est souhaitable de présenter à l'utilisateur. Ces deux possibilités n'ont pas été spécifiées plus finement.

#### 4.2.1.3. Le magasin

Une liste de couples est proposée à l'utilisateur qui ont la forme [nom du produit, prix unitaire]. L'utilisateur doit pouvoir sélectionner un ou plusieurs de ces objets, ceux-ci venant d'une façon ou d'une autre sur la liste des achats effectués. On indiquera quelle est la somme de départ, la

somme au fur et à mesure des achats et le total des achats. Un objet dont le prix dépasse la somme restante ne peut être sélectionné. Dans ce cas, on expliquera cet état de choses à l'utilisateur en lui indiquant également la somme manquante. En outre, il sera toujours possible d'enlever n'importe quel objet de la liste des achats et également de vider entièrement le "caddie".

#### 4.2.1.4. L'épargne et le budget

Ces fonctions n'ont pas été davantage spécifiées. Cependant, on peut se baser à ce propos sur le logiciel réalisé sur PC dont les spécifications fonctionnelles de la prévision budgétaire et de l'épargne ont été très bien définies par les groupes de travail qui ont collaboré au projet. Si cette partie devait être développée dans les mois à venir, c'est sur base de ces travaux qu'il faudrait partir.

#### 4.2.2. La spécification des interfaces abstraites (la tâche)

Il s'agit donc ici de définir la tâche, c'est-à-dire les objets abstraits interactifs nécessaires à la réalisation des spécifications fonctionnelles. En d'autres mots, nous devons spécifier les composantes du dialogue (quels sont les messages à afficher et les données à saisir) sans indiquer comment, sous quelle forme ces composantes seront présentées à l'utilisateur.

Nous nous limiterons ici à détailler le sous-système utile réalisé, celui de la tenue des comptes. Il faut noter que les spécifications ne sont pas écrites dans un langage formalisé.

La fonction "tenue des comptes" devient donc une application à part entière pour les raisons déjà évoquées plus haut. Il s'agira donc d'implémenter cette fonction, plus certaines spécifications extra-fonctions, à savoir l'identification de l'utilisateur et le sauvetage des diverses informations au sujet de la session en cours (le choix entre les cinq fonctions - tenue des comptes, catalogue, achats, épargne et budget - ne sera pas implémenté étant donné que seule une de ces fonctions est développée). On peut tout de suite noter que le sauvetage des données sera tout à fait transparent vis-à-vis de l'utilisateur handicapé. En effet,

ces renseignements sont soit internes car utilisés par le programme lui-même ou destinés aux éducateurs et concepteurs (voir le chapitre 6 au sujet de la trace des manipulations). Ceci signifie donc qu'aucun objet d'interface ne correspondra à cette partie des spécifications.

Pour bien comprendre le déroulement du programme de tenue des comptes au point de vue interaction, on peut présenter l'algorithme suivant :

- > Identification de l'utilisateur (1)
- > Saisie de la somme initiale (2)
- > Tant que l'utilisateur ne désire pas quitter le programme
  - > Présentation du choix entre les différentes options (3)  
(recette, dépense, état du porte-monnaie, FIN)
  - > Tant que l'option choisie n'est pas celle de fin
    - > Activation de l'option choisie (4)
    - > Présentation du choix entre les différentes options (3)
  - > Présentation du (dés)équilibre des comptes et présentation du choix entre quitter le programme ou non (5).

Voyons maintenant en détail quels sont les objets abstraits (ou messages interactifs) nécessaires pour afficher les informations à l'utilisateur et saisir les données qui permettront le déroulement du programme, et voyons comment s'organise effectivement le dialogue, dans chaque fonction de notre sous-système utile.

#### 1) L'identification de l'utilisateur :

Il s'agit donc ici de :

- indiquer à l'utilisateur qu'il doit communiquer son prénom.
- saisir la réponse de l'utilisateur + assurer un feed-back.
- demander confirmation du prénom communiqué.
- saisir la (non) confirmation + feed-back (donc si le prénom n'est pas confirmé, cela implique que l'on recommence au début).

#### 2) Saisie de la somme initiale :

Il s'agit donc de l'argent dont dispose réellement l'utilisateur.

- indiquer à l'utilisateur qu'il doit communiquer la somme qu'il a dans son porte-monnaie.
- saisir le montant et assurer le feed-back.
- demander confirmation de la somme communiquée.
- saisir la (non) confirmation + feed-back.

### 3) Présentation du choix entre les différentes options

Donc, à ce niveau, on ne fait aucune hypothèse sur la façon dont sera présenté ce choix (nous verrons, plus tard, qu'il s'agira d'un menu).

- indiquer à l'utilisateur qu'il doit choisir une option.
- proposer à l'utilisateur 4 options : enregistrer une recette, enregistrer une dépense, visualiser ce que l'on a dans le porte-monnaie et la différence avec le solde des comptes, quitter le programme.
- saisir le choix de l'utilisateur + feed-back.

### 4) Les options

#### 1° Enregistrer une recette

Cela se fait en trois étapes :

- \* saisie de l'origine de la recette (postes budgétaires) :
  - indiquer à l'utilisateur qu'il doit communiquer l'origine de la recette.
  - proposer une liste de postes représentant les origines possibles d'une recette.
  - saisir le poste choisi par l'utilisateur.
  - demander confirmation.
  - saisir la (non) confirmation + feed-back.
- \* saisie du moment de la recette
  - indiquer à l'utilisateur qu'il doit communiquer quand il a reçu cet argent.
  - proposer les différents moments possibles : soit les 7 jours de la semaine, soit les 3 parties de la journée, (selon le choix opéré dans les paramètres).
  - saisir le moment choisi par l'utilisateur.



- demander confirmation du moment.
- saisir la (non) confirmation + feed-back.
- \* saisie du montant de la recette : cette étape ressemble fortement à la saisie de la somme initiale, on trouve les éléments suivants :
  - indiquer à l'utilisateur qu'il doit communiquer le montant de la recette.
  - saisir le montant de la recette et assurer un feed-back.
  - demander confirmation de cette somme
  - saisir la (non) confirmation + feed-back.
  - permettre de quitter cette option sans avoir réellement enregistré quoi que ce soit (ceci dans le cas où l'utilisateur a choisi cette option par erreur).

## 2° Enregistrer une dépense

Cette option se comporte de la même façon que l'enregistrement d'une recette, les postes budgétaires étant évidemment différents, ainsi que les messages qui ont maintenant rapport à une dépense et plus à une recette.

## 3° Etat du porte-monnaie

Il s'agit de :

- indiquer la somme qui se trouve dans le porte-monnaie et qui a été communiquée par l'utilisateur au début du déroulement du programme (somme initiale).
- indiquer le jour actuel de la semaine (ou l'heure du jour si le cycle est d'un jour).
- indiquer la différence éventuelle entre le solde des comptes et la réalité, et indiquer la nature de cette différence : des dépenses n'ont pas été enregistrées ou bien des recettes.

## 5) Activation de l'option FIN (sortie du programme)

- demander la confirmation de la demande de fin.
- saisir la (non) confirmation + feed-back.
- indiquer s'il existe un déséquilibre entre la somme réelle

(initiale) et l'état des comptes, et si c'est le cas, spécifier s'il s'agit d'un oubli de recettes ou de dépenses.

- proposer à l'utilisateur le choix entre revenir au menu pour combler les oublis éventuels ou quitter le programme malgré un possible déséquilibre.
- saisir le choix de l'utilisateur + feed-back.

#### 4.2.3. La spécification des interfaces concrètes (la présentation)

Le but de cette section est donc de définir la présentation, selon la terminologie de J.C. Ruche, c'est-à-dire déterminer la forme effective que vont prendre les divers éléments de la tâche. Il est clair que, lorsque l'on développe des applications interactives pour handicapés mentaux, le travail de spécification est très important. Si la présentation laisse à désirer, le programme, même si les fonctionnalités sont très bien définies, perdra toute son utilité.

Nous avons déjà parlé de l'hétérogénéité de la population handicapée mentale et la difficulté de classification de ces personnes. Cela nous a amené à dire que c'est presque une méthode "cas par cas" qu'il faut adopter lorsque l'on désire inculquer une connaissance quelconque aux handicapés mentaux. Nous avons essayé de respecter cette façon de voir en proposant plusieurs dialogues, plusieurs objets concrets pour un même élément de la tâche. Ceci est rendu possible par le programme auxiliaire de paramétrisation qui permet de choisir des éléments de dialogue pour chaque utilisateur. Cela a l'avantage de permettre une adaptation du dialogue la plus grande possible à chacun des handicapés utilisant le programme.

Reprenons donc les éléments de la tâche et voyons les objets interactifs concrets qui y correspondent.

## 1) L'identification de l'utilisateur

Un seul type de dialogue est prévu pour cette étape et c'est bien normal puisque pour retrouver les paramètres de dialogue d'un utilisateur, il faut d'abord l'avoir identifié.

- \* Pour indiquer à l'utilisateur qu'il doit donner son nom, la suite de caractères "Quel est votre nom ?" apparaît à l'écran.
- \* La saisie de la réponse se fait via le clavier ; l'utilisateur doit donc taper chaque caractère de son nom, suivi d'un return. Le nom doit comporter de 1 à 19 caractères. Le feed-back est assuré par l'affichage automatique de chaque caractère (sauf le return) au fur et à mesure qu'ils sont tapés au clavier. Le feed-back du return (représentant la fin d'entrée du nom) est assuré par le passage automatique à la demande de confirmation.
- \* La demande de confirmation se fait par un requester ou boîte de dialogue modale (voir à ce sujet la section 1.2.2.1.) contenant une phrase et deux boutons et apparaissant à l'écran en-dessous du nom tapé au préalable. Les boutons sont des cases sur lesquelles l'utilisateur peut cliquer avec la souris afin de sélectionner ce qui est représenté dans cette case (il s'agit en fait d'icônes non graphiques, contenant un élément textuel). La phrase est en fait la question "Est-ce bien juste ?" et les deux boutons contiennent respectivement les mots OUI et NON. La saisie de la réponse à cette demande de confirmation se fait évidemment via la souris : l'utilisateur vient placer le suiveur de la souris sur une des deux cases et ensuite appuie sur le bouton gauche de la souris. Le feed-back de cette action dépend de la réponse donnée par l'utilisateur. La réponse NON implique que le requester disparaît ainsi que le nom préalablement tapé ; la question reste indiquant à l'utilisateur qu'il peut taper un nouveau nom. La réponse OUI donne lieu à un effacement total de l'écran et un passage à la saisie de la somme initiale.

## 2) Saisie de la somme initiale

Ici, plusieurs types de dialogue sont prévus pour chaque élément de la tâche.

- \* Indiquer à l'utilisateur qu'il doit communiquer la somme qu'il a actuellement dans son porte-monnaie peut être réalisé :
  - par une phrase s'affichant à l'écran : "Combien avez-vous d'argent ?"
  - et/ou par cette même phrase prononcée par le synthétiseur vocal de l'Amiga.

- \* Pour saisir le montant et assurer un feed-back + demander confirmation de la somme entrée, saisir la réponse et assurer un feed-back, deux représentations sont possibles :

a) Dans la première, la saisie se fait via le clavier. L'utilisateur tape chaque caractère numérique constituant le montant. Les décimales sont permises, pouvant être séparées de la partie entière par un point ou une virgule. Etant donné qu'en Suisse, un montant sans partie décimale est généralement suivi de ",-" ou ".-" (par exemple : 200,- ou 200.-), cette situation est acceptée dans le programme. Tous les autres cas sont exclus et une erreur est traduite par l'effacement automatique de la somme entrée dès que la touche du return est enfoncée (un montant correct est converti en réel).

Le feed-back est automatique en ce sens que chaque caractère tapé est affiché à l'écran. Le feed-back du return de fin d'input est réalisé par le passage à la demande de confirmation de la somme communiquée, ainsi que par la mise à jour du niveau du thermomètre de couleur. Le thermomètre de couleur est un rectangle de 1.5 cm de base et de 8 cm de hauteur qui peut plus ou moins se remplir à partir du bas dans une certaine couleur. Ainsi, si le thermomètre est rempli de moitié et que la valeur totale du thermomètre représente une somme de 1000 Frs, la partie colorée représente un montant de 500 Frs. Nous avons utilisé ce principe de la façon suivante : lorsque l'on saisit la somme initiale, le thermomètre est vide et se remplit de la somme entrée proportionnellement à la valeur représentée par le thermomètre plein. Cette valeur maximale est propre à chaque utilisateur et doit être spécifiée dans le programme auxiliaire de paramétrisation. Ainsi, si la valeur maximale qui est associée à un utilisateur est de 2000 Frs et qu'il déclare avoir dans son porte-monnaie une somme de 500 Frs, le

thermomètre se remplira du quart à partir du bas en couleur bleue. De plus, un repère jaune ira se placer également au quart de la hauteur du thermomètre et restera à cet endroit dans tous les thermomètres qui apparaîtront par la suite. Ce repère représente la somme que l'utilisateur a dans son porte-monnaie, c'est-à-dire la somme initiale.

La demande de confirmation de la somme communiquée se fait par l'apparition d'un requester qui contient deux boutons : l'un s'appelle EFFACE et l'autre FINI. La saisie se fait donc par la souris lorsque l'utilisateur clique sur un de ces deux boutons. Le feed-back dépend du bouton sur lequel le handicapé a cliqué. S'il s'agit du bouton FINI, on passe à la présentation du choix entre les différentes options (voir 3), sinon le requester disparaît, le thermomètre se vide (le repère jaune revenant au bas du thermomètre), la somme préalablement tapée s'efface et la question reste, invitant l'utilisateur à communiquer une nouvelle somme initiale.

- b) La seconde représentation est tout à fait graphique. Elle constitue une saisie pas à pas de la somme contenue dans le porte-monnaie. Ainsi, dans la partie gauche de l'écran est affiché l'ensemble des représentations symboliques des billets et pièces existant dans la monnaie utilisée (belge ou suisse). Un billet est représenté par un rectangle blanc à bord bleu dans lequel est inscrit le montant du billet et une pièce par un cercle blanc à bord bleu dans lequel est inscrit le montant de la pièce. Ces différents billets et pièces sont des boutons, c'est-à-dire que ce sont des objets interactifs graphiques que l'utilisateur peut sélectionner grâce à la souris. La personne handicapée va donc sortir le contenu de son porte-monnaie et, pour chaque billet et chaque pièce, elle va cliquer sur la représentation symbolique correspondante. On voit donc que le choix de cette présentation implique que la partie de l'interface abstraite que nous avons désignée par "saisir le montant de la somme initiale" est en fait constitué de sous-étapes :

- saisir un billet ou une pièce
- assurer un feed-back de l'unité monétaire saisie.

La saisie d'un billet ou d'une pièce se fait, comme nous l'avons indiqué, par l'intermédiaire de la souris. Il y a plusieurs feed-back :

\* la même représentation du billet ou de la pièce, mais de taille inférieure, apparaît dans la partie droite de l'écran. Lorsque ce billet (ou cette pièce) est sélectionné une deuxième fois, un deuxième billet (ou pièce) de petite taille apparaît à droite du premier et ainsi de suite. Cependant, étant donné la taille limitée de l'écran, seuls 5 occurrences d'un même billet et 7 occurrences d'une même pièce peuvent apparaître à l'écran. Au delà de ce nombre, ce type de feed-back n'est plus assuré (cette limitation est donc due aux dimensions de l'écran, mais pour un certain nombre d'utilisateurs ce n'est pas trop grave ; ceux qui ne savent pas compter ou pas compter jusque 5).

\* le total sous forme numérique apparaît en bas à droite de l'écran. Il s'agit en fait de la phrase "TOTAL \$\$\$ FRS", le total étant mis à jour au fur et à mesure que les billets et pièces sont sélectionnés.

\* la valeur de la pièce ou du billet sélectionné est prononcée par la synthèse vocale (rappelons que le dernier message prononcé peut toujours être réécouté en sélectionnant l'icône représentant une oreille).

\* le thermomètre de couleur constitue le dernier feed-back. La sélection d'un billet ou d'une pièce fait proportionnellement croître d'autant la partie bleue du thermomètre. Le repère jaune suit la même évolution.

Cette représentation a impliqué également la présence de trois autres objets interactifs ne correspondant pas à des messages interactifs.

-> Il s'agit d'abord d'un bouton nommé "FINI" jouant le rôle de return de la première présentation. Lorsque l'utilisateur a communiqué tous ses billets et pièces, il le sélectionnera, ce qui donnera lieu à l'apparition d'un requester qui demandera "Avez-vous fini?" et qui contiendra un bouton "OUI" et un "NON". Le bouton "OUI" provoque le passage à la présentation du choix entre les différentes options (nous verrons qu'il sera présenté sous forme d'un menu graphique), tandis que le bouton "NON" permet de continuer la saisie comme si rien ne s'était produit.

-> Le deuxième objet est un bouton appelé "EFFACE" permettant d'effacer la dernière pièce ou le dernier billet entré. Lorsque l'utilisateur se rend compte qu'il vient de commettre une erreur

(notamment grâce aux différents feed-back qui ont suivi sa dernière action), il clique sur ce bouton. Ici aussi apparaît un requester demandant la confirmation de l'ordre d'effaçage, il pose la question suivante : "Voulez-vous effacer?", et contient aussi un bouton "OUI" et bouton "NON". Le bouton "OUI" aura pour effet d'effacer le dernier petit billet (ou pièce) affiché comme feed-back dans la partie droite de l'écran, de remettre le total à jour, ainsi que le thermomètre de couleur (la partie bleue décroît proportionnellement) et permet de continuer la saisie. Le bouton "NON" a pour effet de ne pas effacer le dernier billet ou pièce et de continuer la saisie comme si rien ne s'était passé.

Il faut noter que cet effacement n'est possible qu'à un niveau ; ainsi si on appuie deux fois successivement sur le bouton "EFFACE", le deuxième ordre d'effacement n'aura aucun effet.

- > Enfin, un troisième bouton nommé "RECOMMENCER" permet d'effacer tous les billets et pièces qui ont été entrés, de remettre le total et le thermomètre à 0. Ici aussi une confirmation est demandée avant de réaliser effectivement l'opération, elle a la même forme que les précédentes.

Avec ces trois objets supplémentaires (propres à cette présentation), une confirmation de la somme totale entrée n'est plus nécessaire. Elle est implicite, vu que l'utilisateur a pu se corriger au fur et à mesure de la saisie, et qu'après avoir appuyé sur "FINI" il doit confirmer cette demande.

### 3) Présentation des différentes options

Dans ce cas aussi, un seul type de dialogue est possible et il s'agit d'un menu graphique. Ainsi, les quatre options recette, dépense, état du porte-monnaie et FIN sont représentées par quatre grosses icônes. La saisie du choix opéré se fait par clic souris sur l'icône représentant l'option choisie. Le feed-back est en fait constitué par l'activation de la fonction en question. On indique à l'utilisateur qu'il doit choisir une option par une phrase "Choisissez une activité" qui peut être affichée à l'écran et peut être aussi prononcée par la synthèse vocale.

#### 4) Les options

##### A) Communication d'une recette

- \* La saisie de l'origine de la recette prend la forme d'un écran contenant le message "D'où vient cette recette ?" qui peut être affiché et peut être également prononcé et un ensemble de quatre icônes représentant les origines possibles : parents, argent de poche, travail, économie. Le feed-back est assuré par un requester contenant une phrase affichée et/ou prononcée : "Cela vient bien de ..." actualisé au poste choisi, ainsi que deux boutons OUI et NON, le OUI faisant passer à l'étape de saisie du moment de la recette et le NON effaçant le requester et permettant une nouvelle sélection.
- \* La saisie du moment de la dépense dépend du cycle choisi dans le programme de paramétrisation (une semaine ou un jour).

##### Cycle d'une journée

Une question affichée et qui peut être aussi prononcée "Quand avez-vous fait cette recette ?" indique à l'utilisateur qu'il doit choisir parmi les trois moments de la journée qui sont le matin, l'après-midi et le soir. Ces trois moments sont représentés par trois icônes sur lesquelles l'utilisateur clique pour poser son choix. Dès qu'une icône est sélectionnée, un requester apparaît demandant confirmation du moment choisi. Cela se fait de la même façon que pour le requester demandant confirmation du poste choisi.

##### Cycle d'une semaine

La même question que pour le cycle d'un jour est proposée. Apparaissent également sept icônes représentant les sept jours de la semaine. Si le cycle commence le lundi et que l'on est jeudi, il est clair que la personne handicapée ne peut avoir fait la recette en question que le lundi, mardi, mercredi ou jeudi. Dès lors, si l'utilisateur clique sur une icône représentant un jour ultérieur au jeudi, cette action n'a aucun effet. Cliquer sur un jour antérieur ou égal au jeudi donne lieu à l'apparition d'un requester de confirmation. Il faut également noter que, dans ce cas, les icônes représentant les jours du lundi au mercredi sont de couleur



verte, indiquant que ces jours sont passés, le jeudi est de couleur blanche montrant qu'il s'agit du jour actuel et les jours du vendredi au dimanche sont de couleur rouge pour attirer l'attention sur le fait qu'il faut encore "tenir" ces trois jours avec la somme qui reste. Il faut encore remarquer que le cycle d'une semaine ne doit pas nécessairement commencer le lundi. En effet, il est possible que le jour repère du début de semaine de l'utilisateur handicapé soit le mercredi parce que, par exemple, il reçoit sa paie tous les mercredis. Dans ce cas, la semaine représentée commencera le mercredi, pour autant que l'éducateur l'ait spécifié dans le programme de paramétrisation.

- \* La saisie du montant de la recette présente les mêmes caractéristiques que la saisie de la somme initiale. En effet, quel que soit le type de montant saisi, il s'agit toujours de la saisie d'une somme. Les différences sont les suivantes :

Tout d'abord, il est évident que la phrase proposée est différente. Dans ce cas-ci, la question sera "Combien avez-vous reçu d'argent ?".

Quel que soit le type de présentation choisi, un bouton supplémentaire est nécessaire. En effet, si la saisie de la somme initiale était obligatoire (c'est-à-dire devait nécessairement être effectuée, pour pouvoir assurer un minimum de contrôles), la saisie d'une recette ne l'est pas et il est possible que l'utilisateur ait choisi cette option par erreur. Ce bouton, appelé QUITTER, permet à l'utilisateur de revenir au menu sans avoir enregistré quoi que ce soit. Un requester de confirmation permet de confirmer cet ordre (uniquement dans le cas de la présentation graphique).

Dans la présentation graphique, le bouton RECOMMENCER n'existe plus, principalement pour des questions de place à l'écran. De plus, étant donné l'existence du bouton QUITTER, ce bouton n'est plus nécessaire puisque si des erreurs ont été commises, il suffit de quitter et de re-sélectionner cette même option.

Dans les deux types de présentation, une icône rappelant que l'on se trouve dans l'option recette est affichée à l'écran (il s'agit de la même icône que celle affichée dans le menu). Ceci est dû au fait que toutes les saisies d'une somme se ressemblent. Cette icône ne correspond donc à aucun objet interactif abstrait.

Enfin, le fonctionnement du thermomètre est différent par rapport à la saisie de la somme initiale. Avant que le montant de la recette soit effectivement communiqué, une partie du thermomètre est coloré de bleu représentant la valeur actuelle de l'état des comptes, c'est-à-dire le solde de la session précédente + le total des recettes déjà communiquées - le total des dépenses déjà entrées. Le repère jaune se trouve toujours au même niveau qu'après la saisie de la somme initiale. Lorsque cette nouvelle recette est communiquée, le niveau augmente proportionnellement en couleur verte, ceci afin que la personne handicapée se rende compte de ce que vaut cette nouvelle recette par rapport à ce qu'elle a déjà. Ce n'est que lorsque l'utilisateur aura cliqué sur le bouton FINI et confirmé son ordre que cette partie verte redeviendra bleue, représentant alors le nouvel état des comptes. Le but du jeu est d'essayer que le niveau bleu du thermomètre arrive exactement au repère jaune. Si tel est le cas cela signifie que les comptes sont à jour, qu'ils sont le reflet exact de la réalité.

#### B) Communication d'une dépense

La communication d'une dépense se passe de la même façon que la saisie d'une recette, à noter que :

Pour la saisie du poste de la dépense, ceux-ci sont de nature différentes. Il s'agira toujours d'icônes, mais elles représenteront cette fois de la nourriture, du tabac, des habits, des transports, des magazines, la tirelire, les loisirs et des choses diverses.

Le fonctionnement du thermomètre est évidemment un peu différent. Lorsque le montant de la dépense est entrée (ou à chaque pas pour la présentation graphique), le niveau bleu diminue en rouge (encore une fois pour montrer la proportion de la dépense par rapport à l'état des comptes), et quand l'utilisateur a cliqué sur FINI et confirmé sa demande, cette partie rouge disparaît définitivement pour indiquer le nouvel état des comptes.

Ici, c'est évidemment l'icône de la dépense qui rappelle l'option dans laquelle on se trouve.

### C) Etat du porte-monnaie

Il y a deux présentations différentes, mais la liberté laissée à l'éducateur n'est pas totale, comme nous l'expliquerons un peu plus loin.

#### a) Présentation alphanumérique

\* Dans cette présentation, c'est une phrase qui rappelle à l'utilisateur combien il a dans son porte-monnaie. Cette phrase est la suivante "Il vous reste ... Francs" actualisée au montant de la somme initiale. Elle est affichée et peut être également prononcée.

\* Il faut indiquer à quel moment nous sommes dans le cycle de temps de l'utilisateur :

##### Cycle d'une semaine

la présentation de la semaine suit la même logique que lorsqu'on saisit le jour d'une recette ou d'une dépense. Ainsi, les jours précédant le jour actuel sont verts, le jour actuel est blanc et les jours à venir sont rouges (ceci donne donc une idée de l'argent qu'il reste par rapport au temps qu'il reste avant la fin du cycle).

##### Cycle d'une journée

Il s'agit en fait d'une horloge graphique en ce sens qu'il s'agit d'un rectangle posé sur sa base la plus grande et divisé en 24 cases représentant les 24 heures du jour. Tous les quarts d'heures, une tige blanche vient remplir une partie d'une case. Lorsque la case est complètement remplie (par quatre tiges), elle devient toute rouge. Ainsi, s'il est 14h50, 14 cases seront remplies et rouges et la case suivante sera remplie au trois quarts, mais de blanc.

\* Le thermomètre représente la situation actuelle de l'état des comptes. Trois situations sont possibles :

- l'état des comptes est égal à la somme initiale, auquel cas le niveau bleu atteint le repère jaune sans le dépasser
- l'état des comptes est inférieur à la somme initiale, ce qui est traduit par le fait que le niveau bleu n'atteint pas le repère jaune. La différence est remplie de rouge
- l'état des comptes est supérieur à la somme initiale, ce qui est

représenté par le fait que le niveau bleu dépasse le repère jaune. La différence est remplie de rouge.

- \* Un bouton FINI permet de passer à la suite. Encore une fois, ce bouton ne correspond à aucun objet interactif abstrait, mais vient du fait que nous voulions surtout attirer l'attention sur le fait que les comptes correspondent ou pas à la réalité.
- \* Lorsque le bouton FINI est cliqué, et si les comptes ne sont pas à jour, un requester apparaît contenant une phrase qui dépend de la nature du déséquilibre. Si l'état du porte-monnaie est inférieur à la somme initiale, le message est "Il vous manque ... Francs de recettes", actualisé à la différence calculée entre la somme initiale et l'état des comptes. Si c'est l'inverse, la phrase sera "Il vous manque ... Francs de dépenses". Ce requester contient également un bouton, appelé lui aussi FINI, qui permet de sortir de cette option et de revenir au menu des différentes options.

#### b) Présentation graphique

Cela fonctionne de la même façon, à part que la présentation de la somme initiale est différente. Dans ce cas-ci, la somme initiale apparaît sous forme de représentation symbolique des billets et des pièces qui constituent cette somme. Ainsi, si l'utilisateur a spécifié au départ qu'il avait dans son porte-monnaie une somme de 380 Francs en cliquant trois fois sur le billet de cent francs et quatre fois sur la pièce de vingt francs, ce sont ces sept icônes que l'on verra apparaître à l'écran.

Bien entendu, si l'utilisateur a spécifié sa somme initiale non pas en cliquant sur des billets et des pièces, mais en tapant le montant au clavier, il est impossible au programme de déterminer de quelles unités est composé cette somme. Ainsi, un montant de 86 Francs peut provenir de quatre pièces de vingt francs et de six pièces d'un franc ou d'une pièce de cinquante francs, d'une pièce de vingt francs, de trois pièces de cinq francs et d'une pièce d'un franc (ou encore d'autres possibilités). Dès lors, même si l'éducateur a spécifié, dans le programme de

paramétrisation, qu'il désirait la saisie de la somme initiale par clavier et le rappel de celle-ci par billets et pièces, la présentation se fera sous forme alphanumérique.

#### 5) Activation de l'option FIN (sortie du programme)

Il faut tout d'abord, lorsque cette option est sélectionnée, demander une confirmation au cas où il s'agirait d'une erreur de manipulation. Cette confirmation se fait par l'apparition d'un requester comme nous en avons déjà souvent parlé.

L'étape suivante consiste à montrer si les comptes sont à jour ou non, c'est-à-dire si la somme initiale est égale à l'état des comptes :

##### A) Comptes à jour

Si la somme initiale est égale à l'état des comptes, le thermomètre apparaît présentant la situation actuelle, dans laquelle le niveau bleu a atteint le repère jaune sans le dépasser. De plus, la phrase suivante apparaît à l'écran : "Bravo, les comptes sont à jour". Enfin, un dessin apparaît représentant une balance à deux plateaux, l'un des deux portant la lettre R (recette) et l'autre la lettre D (dépense) - ces lettres étant de couleur bleue. Cette balance est en équilibre.

##### B) Manque de recettes.

Cette fois, l'état des comptes est inférieur à la somme initiale, signifiant qu'il manque dans les comptes une certaine somme d'argent.

Le thermomètre présente la situation actuelle dans laquelle le niveau bleu n'atteint pas le repère jaune. La différence est remplie de couleur rouge.

Dans ce cas, la phrase qui apparaît est la suivante : "Il vous manque ... Francs de recettes".

Cette fois, la balance est plus lourde du côté des dépenses, la lettre R est de couleur rouge et une flèche attire l'attention sur elle (indiquant qu'il faut encore des recettes).

De plus, l'icône des recettes est affichée (la même que dans le menu et dans la saisie du montant d'une recette) pour indiquer que, si l'utilisateur revient au menu, il devrait prendre cette option-là.

QUEL EST VOTRE NOM ?

Alexandre

EST-CE BIEN JUSTE ?

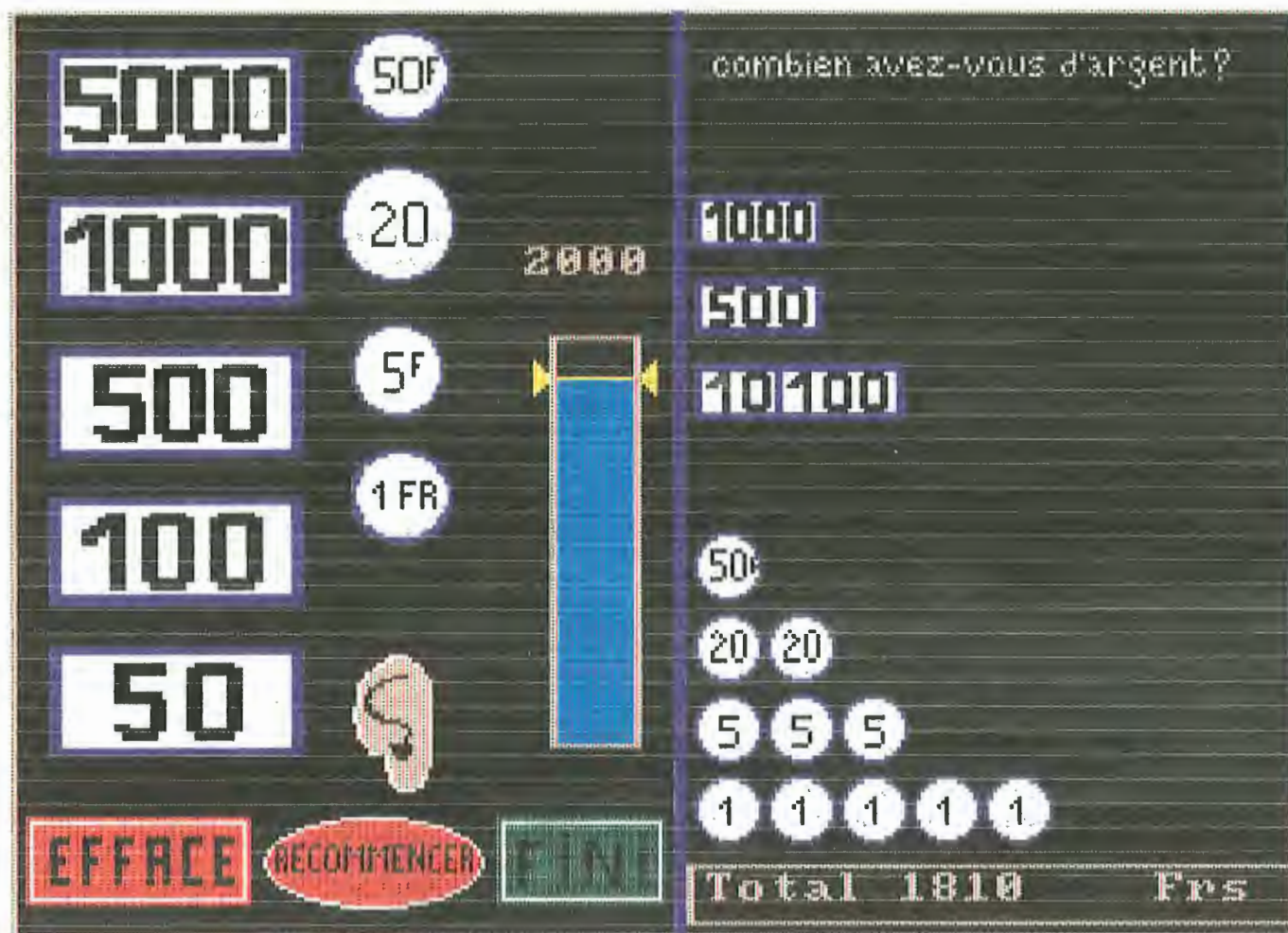
OUI

NON

#### Ecran de saisie du nom

Cet écran permet à l'utilisateur de communiquer son nom au programme. Un requester lui demande confirmation du nom précédemment tapé au clavier.





### Ecran de saisie de la somme initiale (par souris)

L'utilisateur est donc en train de communiquer la somme qu'il a dans son porte-monnaie en ce début de session.




#### Ecran du menu

Ce menu graphique demande à l'utilisateur de choisir parmi les quatre activités : enregistrement d'une recette, d'une dépense, consultation du porte-monnaie et FIN.



D'où vient cette recette ?

PARENTS 	ARGENT DE POCHE 	TRAVAIL 	TIRE LIÈGE 
--	--	--	---

C'est bien l'argent de poche ?

Ecran de saisie du poste de recette

L'utilisateur spécifie par cet écran le poste d'origine de la recette. Un requester demande confirmation du poste choisi.

Quel jour avez-vous fait cette recette?

☒ LUNDI ☐ MARDI ☐ MERCREDI ☐ JEUDI ☐ VENDREDI ☐ SAMEDI ☐ DIMANCHE

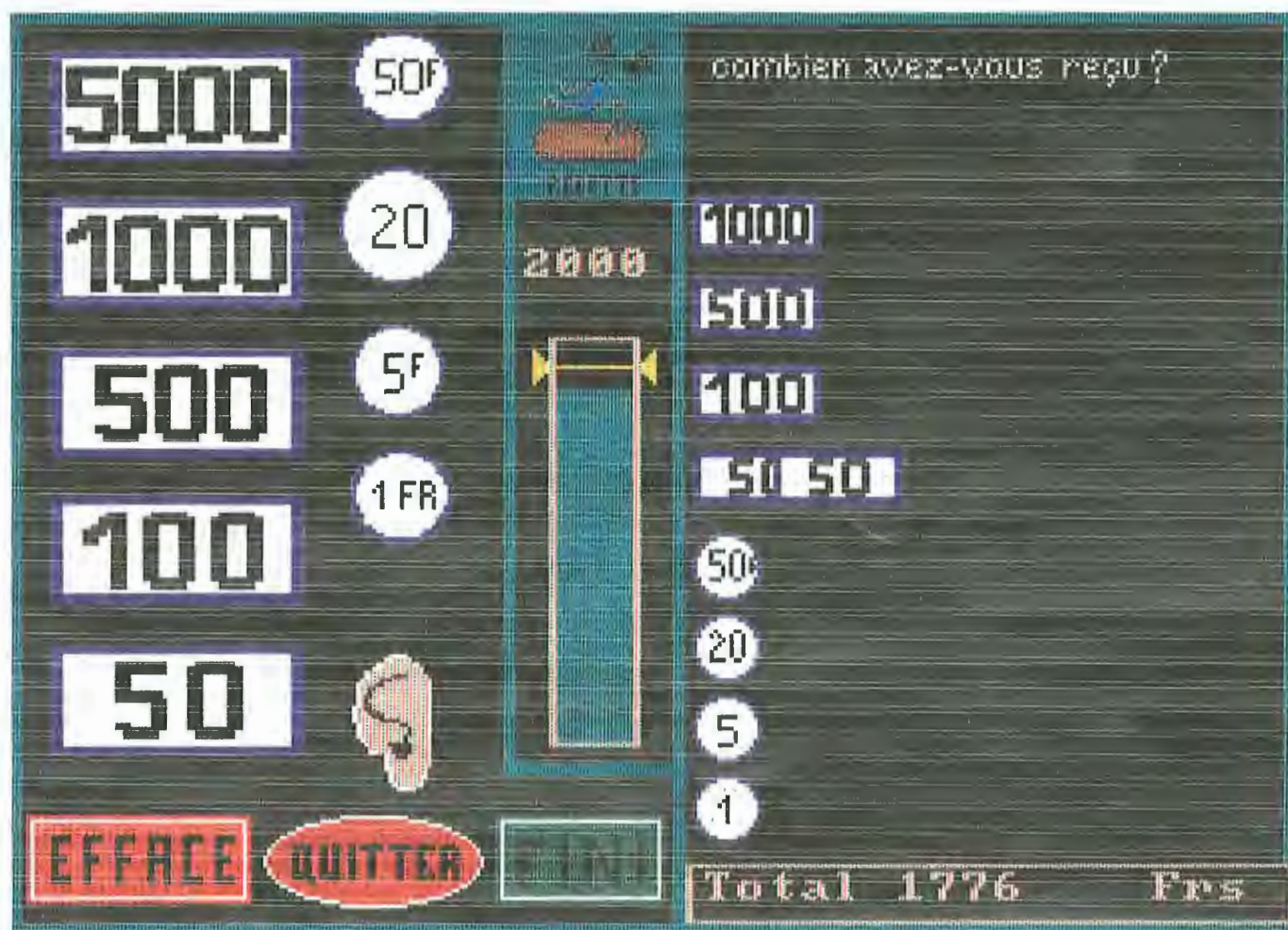
C'est bien le Mardi ?

☐ OUI ☐ NON

Saisie du jour (d'une dépense ou d'une recette)

Cet écran demande au handicapé de spécifier quel jour il a fait sa dépense ou sa recette. Un requester demande confirmation du jour choisi.





### Ecran de saisie d'une recette (par souris)

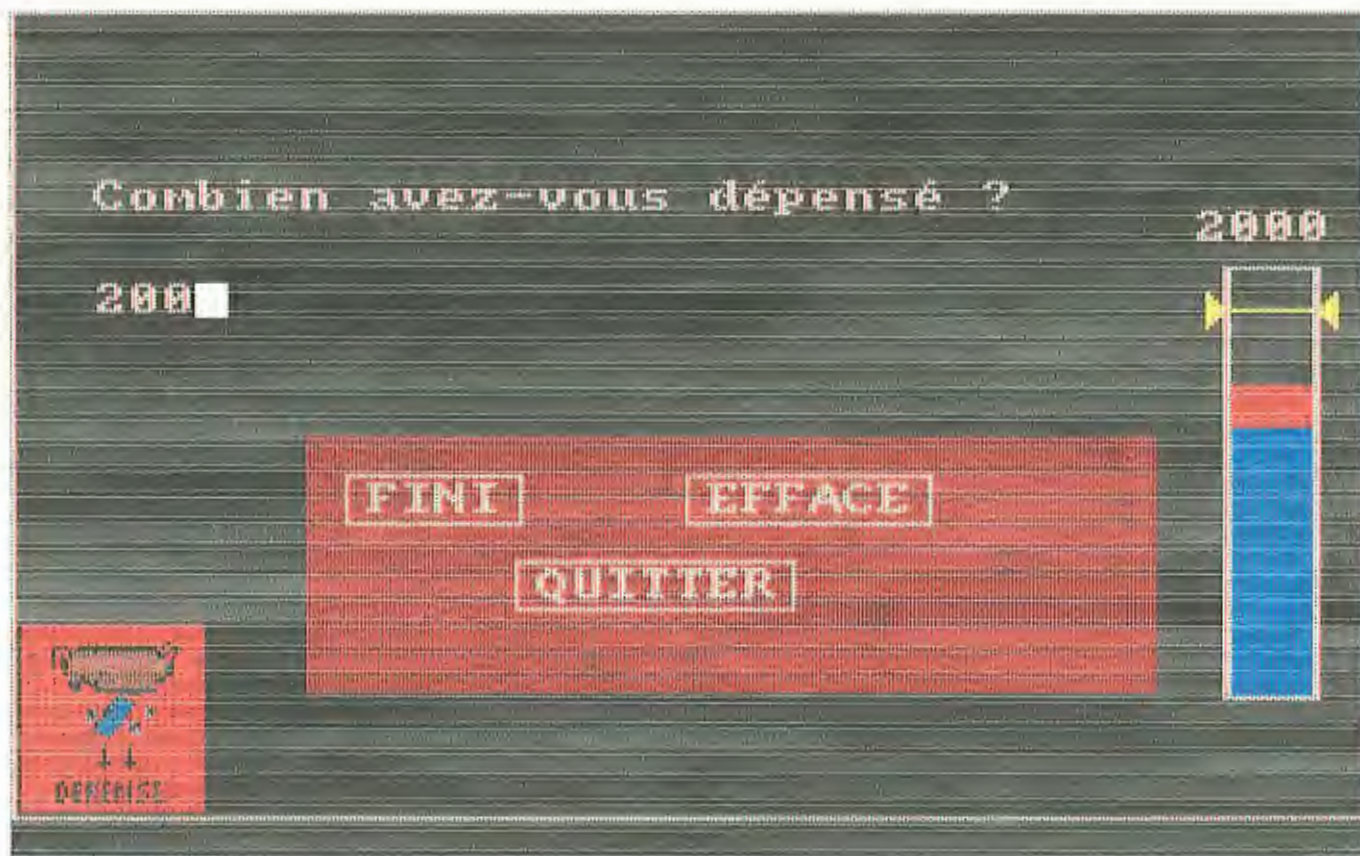
Cet écran permet à l'utilisateur de spécifier pas à pas le montant d'une recette. Le même écran existe pour la saisie d'une dépense, à part que le thermomètre fonctionne différemment et que l'icône représentant l'activité de saisie de dépense est différente (voir saisie d'une dépense par clavier).



#### Ecran de saisie du poste de dépense

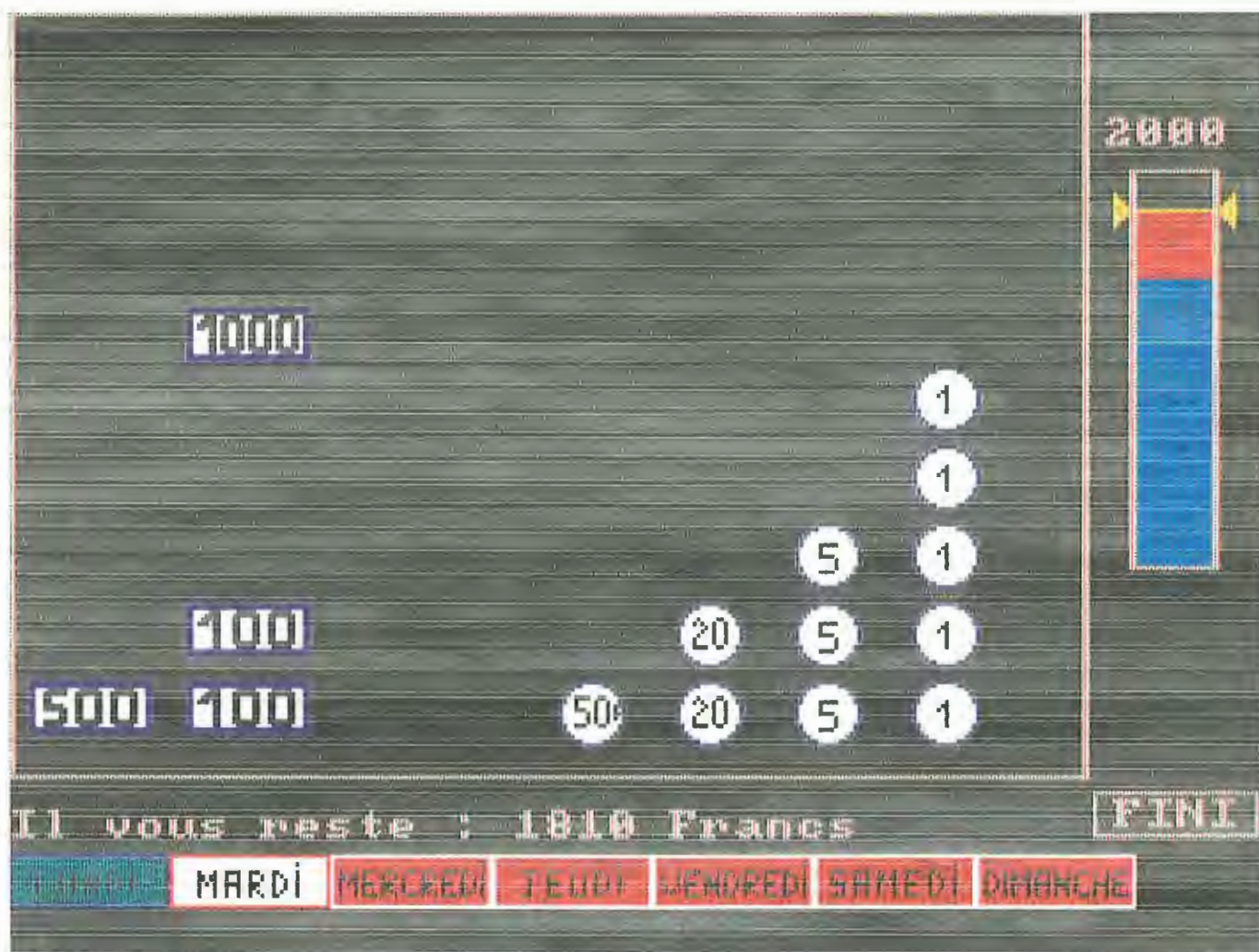
La personne handicapée spécifie le poste budgétaire pour lequel elle a effectué la dépense qu'elle désire communiquer au programme. Un requester demande confirmation du poste choisi.





#### Ecran de saisie d'une dépense (par clavier)

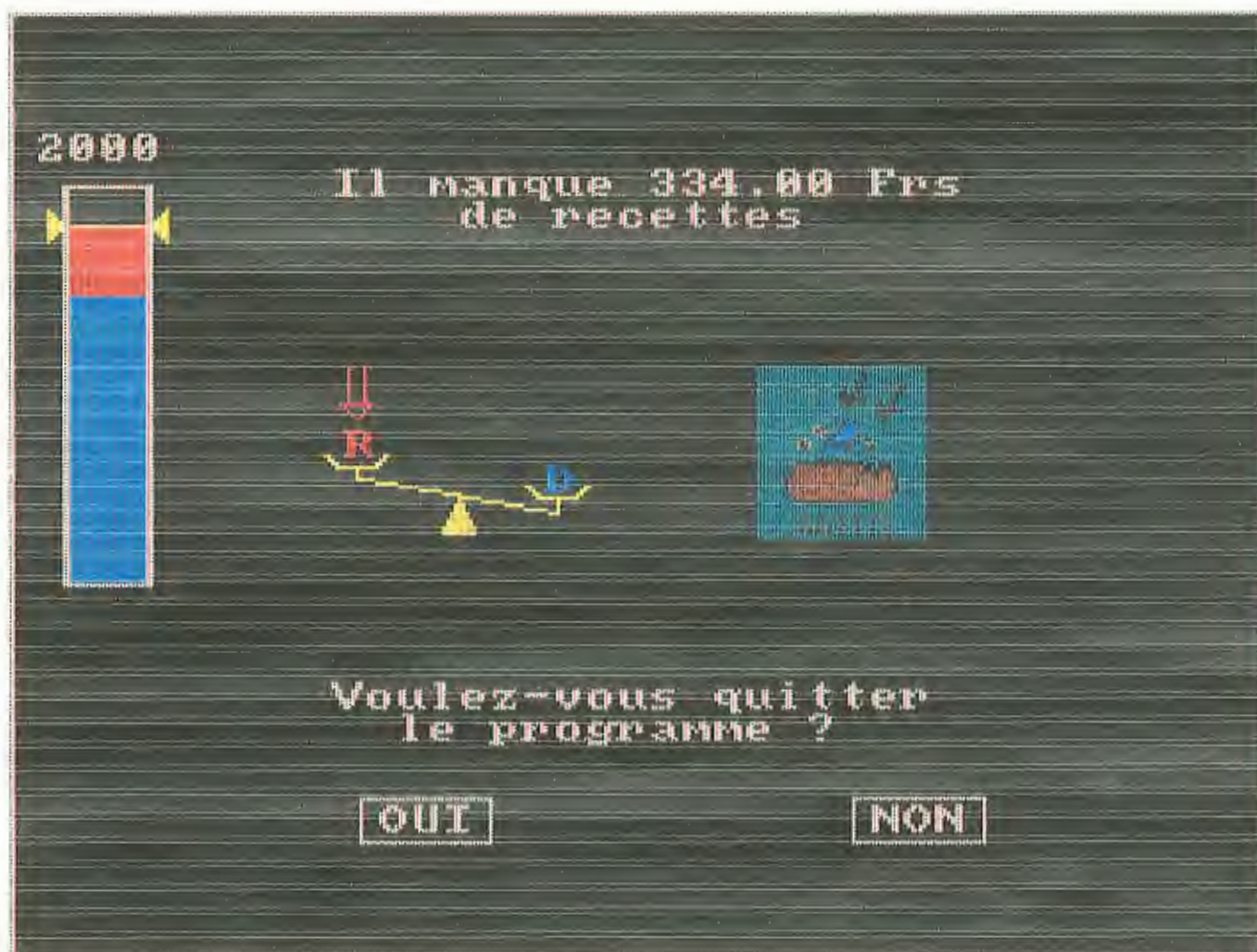
L'utilisateur a communiqué le montant d'une dépense qu'il a préalablement effectuée. Un requester lui demande confirmation de la somme tapée au clavier. Cet écran est le même pour la saisie de la somme initiale et la saisie d'une recette, à part quelques caractéristiques propres à ces activités.



Ecran de consultation du porte-monnaie (graphique)

Cet écran rappelle à la personne handicapée quelle somme elle a actuellement dans son porte-monnaie.





#### Ecran de FIN

Lorsque l'utilisateur désire quitter le programme, cet écran lui indique s'il a ou n'a pas oublié de communiquer des recettes ou des dépenses.

### C) Manque de dépenses.

Cette fois, l'état des comptes est supérieur à la somme initiale, signifiant qu'il y a trop d'argent dans les comptes.

Le thermomètre présente la situation actuelle dans laquelle le niveau bleu dépasse le repère jaune. La différence est remplie de couleur rouge.

Dans ce cas, la phrase qui apparaît est la suivante : "Il vous manque ... Francs de dépenses".

Cette fois, la balance est plus lourde du côté des recettes, la lettre D est de couleur rouge et la flèche attire l'attention sur elle (indiquant qu'il faut encore des dépenses).

De plus, l'icône des dépenses est affichée (la même que dans le menu et dans la saisie du montant d'une dépense) pour indiquer que, si l'utilisateur revient au menu, il devrait prendre cette option-là.

## 4.3. Les spécifications du programme de paramétrisation

### 4.3.1. Remarques préliminaires

Nous avons donc déjà annoncé que le programme d'aide à la tenue des comptes proposerait différents éléments de dialogue (pour un même jeu de fonctionnalités) qui pourraient être combinés par l'éducateur en fonction de la personne destinée à l'utiliser. L'éducateur fera ce choix individualisé grâce au programme de paramétrisation. C'est-à-dire que ce programme permet de dire, pour un individu, les éléments de dialogue que l'on veut ou pas. Précisons tout de suite d'une part, que si aucun jeu de paramètres n'est "créé" pour un utilisateur, il aura le "dialogue par défaut" lors de toute utilisation du programme de comptes, et d'autre part que lorsqu'un éducateur choisit des paramètres, il ne doit plus refaire cette initialisation à chaque session de travail de la personne pour qui ces paramètres sont destinés (il ne devra le refaire que s'il désire changer certains paramètres). Donc en pratique, une initialisation devrait être faite par personne, puis au besoin, si des lacunes apparaissaient, il pourrait être nécessaire de modifier certains paramètres.



En tout cas, il est clair que ce programme sera utilisé uniquement par les éducateurs, qui sont proches des personnes appelées à utiliser le programme de comptes.

#### 4.3.2. Deux aspects contradictoires

Si un logiciel est très paramétré, il sera sans doute très ouvert, mais risque de devenir "lourd" à initialiser. Dans le cas contraire, l'initialisation est simple et courte, mais le logiciel risque d'être beaucoup plus fermé. Il faut donc essayer de concilier ces deux aspects contradictoires. L'expérience tend à prouver que les paramètres sont peu utilisés, toutefois comme l'objectif est aussi d'être adapté au plus grand nombre possible d'utilisateurs handicapés, nous avons essayé de réaliser un programme de paramétrisation le plus court et le plus facile de manipulation possible.

Le programme demande un minimum de frappe au clavier et utilise la souris et la manipulation directe. Pour chaque paramètre, l'éducateur clique sur la case contenant l'alternative qu'il veut.

Voyons rapidement ces paramètres, ainsi que leurs effets sur le déroulement du programme de tenue des comptes.

#### 4.3.3. Les fonctionnalités du programme

Le programme permet donc d'enregistrer une liste de choix, concernant le dialogue désiré pour une personne, voyons ces choix en détail.

Tout d'abord il faut que l'éducateur donne le nom de l'utilisateur pour lequel les paramètres seront valables, nom qui sera utilisé par le handicapé lui-même dans le programme de comptes proprement dit.

Ensuite, s'il le veut, il peut donner une somme qui s'approche du maximum d'argent que cette personne peut avoir durant un cycle (le laps de temps sur lequel il faut gérer son argent). En fait cette somme sert uniquement au "thermomètre" de couleur qui augmente et diminue selon qu'on enregistre des dépenses et des recettes. Il montre donc le solde des comptes. Nous avons vu que ce thermomètre est relatif ; quand il est rempli il représente une somme de référence qui peut être 100 francs ou 1000 ou n'importe quoi. C'est donc cette somme de référence

(qui est représentée par le thermomètre rempli à 100%) qui est demandée à l'éducateur.

Ensuite l'éducateur spécifie la configuration de son ordinateur puisque le programme doit tourner sur trois configurations possibles.

Les deux écrans suivants présentent les paramètres relatifs au dialogue, parmi lesquels l'éducateur devra faire ses choix.

- \* le premier paramètre fixe la monnaie utilisée, puisque le programme propose des interfaces pour la monnaie suisse et la monnaie belge

- \* au sein de chaque fonction, l'ordinateur posera des questions à l'utilisateur, et donnera des feed-back, ce paramètre permet de choisir entre des questions simplement écrites (affichées à l'écran), ou en plus prononcées par la synthèse vocale. Si la deuxième option est choisie, les feed-back acoustiques seront également présents dans le dialogue.

- \* ensuite l'éducateur décide si l'utilisateur communiquera les recettes par le clavier (c'est-à-dire en remplissant un formulaire à un champ) ou par manipulation directe des billets et des pièces avec la souris.

- \* de même pour les dépenses

Ceci signifie, que par exemple les recettes pourront être introduites par manipulation directe et les dépenses par le clavier, parce qu'il est facile de recopier ce qui est imprimé sur un ticket de caisse. Ce sont d'ailleurs ces options qui sont présentées par défaut.

- \* le paramètre suivant fixe la façon dont la somme dont l'utilisateur dispose, lui est rappelée. Elle peut être affichée en chiffres à l'écran, ou uniquement par icônes (représentant les billets et les pièces), ou un de ces deux moyens associé à une prononciation par la synthèse vocale, ou bien les trois ensembles. En plus de cela, il est clair que s'ils ont été choisis, apparaîtront le thermomètre de couleur représentant le solde des comptes, et la règle du temps qui donne une indication sur les jours (ou heures) qu'il reste dans le cycle.

- \* le paramètre suivant permet de dire si le thermomètre de couleur représentant graphiquement le solde des comptes, apparaîtra ou non.

- \* la paramètre suivant permet de dire si la notion de temps devra être utilisée ou pas. Si oui, la règle du temps apparaîtra sur certains écrans. Cette règle représente le cycle sur lequel il faut gérer son argent, et montre graphiquement, quel proportion du cycle est passée et quelle

proportion reste. De plus, si la notion de temps est choisie, cela implique que lors d'enregistrements de recettes et de dépenses, l'utilisateur devra justifier "quand" ces recettes ou dépenses ont été faites. Si le temps n'est pas choisi, ni réglette, ni justification ne seront demandées.

Bien entendu, le fait de pouvoir introduire ces notions temporelles ou non, relève de l'idée de faire évoluer le dialogue en fonction de la personne. Pour certaines il sera heureux de les utiliser tout de suite, pour d'autres il sera peut-être préférable d'avancer graduellement et de commencer par des dialogues sans notions temporelles pour y arriver progressivement. Mais comme nous l'avons vu à la section 4.1.1.2., il n'est pas question de fixer un ordre dans l'apprentissage ; de dire par exemple que telle notion doit être assimilée avant telle autre. Ce que nous pouvons faire, c'est donner la possibilité de combiner les éléments de dialogue, mais sans nous substituer, le moins du monde, au travail du psychologue.

- \* le choix suivant porte sur la justification du poste budgétaire quand on enregistre une recette ou une dépense. De nouveau, on peut imaginer de ne pas le mettre dans un premier temps, puis de l'introduire quand l'utilisateur devient plus expert

- \* le neuvième paramètre détermine si, lorsque l'utilisateur a correctement enregistré toutes ses dépenses et recettes, un récapitulatif de tous ces enregistrements sera affiché à l'écran

- \* la durée du cycle sur lequel l'argent doit être géré, est déterminée par le paramètre suivant. Actuellement on peut apprendre à gérer sur un jour ou sur une semaine. Par conséquent les écrans demandant quand une recette ou une dépense a été faite, varieront en fonction du cycle. Dans un cas l'utilisateur choisira parmi "matin", "après-midi" et "soir", et dans l'autre parmi les sept jours de la semaine.

- \* finalement, si le cycle est d'une semaine, l'éducateur indique quel jour est à considérer comme celui de début de cycle. En effet si la rentrée d'argent principale a lieu un mercredi, le cycle effectif sur lequel il faut gérer son argent, va du mercredi au mardi. Par conséquent, la règle représentant le cycle doit être présentée en fonction de ce cycle effectif.

Comme nous l'avons dit, si l'éducateur ne fixe pas de paramètres pour une personne, lorsque celle-ci utilisera le programme de tenue des

comptes, elle aura le dialogue "par défaut". Voyons les options qu'il propose :

- questions posées par écrit et prononcées par la synthèse vocale
- recettes communiquées par manipulation directe avec la souris
- dépenses communiquées par le clavier
- somme dont dispose l'utilisateur présentée par nombres et par icônes (représentant les billets et pièces)
- présence du thermomètre de couleur
- présence des notions temporelles
- l'utilisateur devra justifier les postes budgétaires correspondant à ses recettes et à ses dépenses
- l'écran qui récapitule les dépenses et recettes sera présenté
- le cycle sera d'une semaine
- il commencera le lundi

#### **4.4. En quoi les règles et principes généraux d'interface sont respectés ?**

Nous allons reprendre rapidement les 8 règles d'or que propose Ben Schneiderman, et les principes de Hansen (que nous avons présentés aux sections 1.2.4.2. et 1.2.4.3.) et nous verrons s'ils ont été respectés et de quelle façon, dans les spécifications du programme, telles qu'elles ont été présentées à la section 4.2..

##### **4.4.1. Respect des 8 règles d'or de B. Schneiderman**

1) Viser l'uniformité : il ne doit pas y avoir d'incohérence entre les différentes interfaces (il faut veiller par exemple à garder des noms identiques pour des fonctions, des concepts identiques).

On peut trouver plusieurs applications de cette règle. Par exemple, quel que soit le dialogue choisi, les différentes fonctionnalités sont toujours présentées par le même menu graphique. De plus, dans chaque fonction, les questions posées sont identiques quels que soient les éléments de dialogue qui ont été choisis par l'éducateur. Les requesters de confirmation restent constants également, la disposition des boutons

"OUI" "NON" aussi (le fait de garder des éléments identiques aux mêmes emplacements est très important, dans l'objectif de faire acquérir des notions).

Nous avons aussi veillé à l'uniformité des couleurs utilisées pour les concepts de recette et de dépense, dans les différents écrans. Par exemple, le vert est associé aux recettes ; l'icône du menu est verte, dans le thermomètre de couleur, c'est en vert qu'apparaissent les recettes entrées, et cela partout où le thermomètre est susceptible d'apparaître. Pour les dépenses, c'est toujours le rouge qui est employé.

## 2) Permettre des raccourcis aux utilisateurs experts.

Cette règle est plus délicate, car elle fait intervenir une notion un peu vague, d'"utilisateur expert". Toutefois, les paramètres permettent de prendre en compte ces "raccourcis". Par exemple s'il est nécessaire à un utilisateur de communiquer le contenu des son porte-monnaie en désignant, avec la souris, un à un les billets et pièces qu'il possède, peut-être qu'à partir d'un certain niveau, il sera possible d'utiliser le dialogue, plus rapide, par le clavier. En ce sens, le passage d'un dialogue par manipulation directe à un dialogue utilisant le clavier, constitue une forme de raccourci offert aux plus experts.

## 3) Offrir un feed-back informatif.

Comme nous l'avons vu, nous avons privilégié le point de vue "feed-back". Notamment dans le cas de dialogue d'introduction d'une somme par manipulation directe avec la souris : un feed-back acoustique est offert après chaque billet ou pièce, un double feed-back graphique avec les petites icônes de billet ou pièce qui apparaissent et le thermomètre qui fluctue pour représenter le solde des comptes. Enfin le total actuel, en chiffres. Puis il faut tenir compte aussi du feed-back que véhicule une demande de confirmation ; par exemple, après avoir introduit un poste budgétaire, le requester qui apparaît reprend le poste choisi et demande s'il s'agit bien du poste désiré. De même, en fin de session de travail, l'écran présentant l'état des comptes avec l'éventuelle différence par rapport à la réalité, constitue en somme un feed-back

global de tout le travail effectué par l'utilisateur.

4) Concevoir des dialogues permettant de découper la tâche de l'utilisateur, afin qu'il ne doive pas fournir un effort de concentration trop long avant de pouvoir clôturer une étape.

Nous avons essayé au maximum, de découper une tâche en sous-tâches plus simples, pour que l'utilisateur puisse toujours avancer par petits pas et se ménager des moments où il peut relâcher sa concentration. Par exemple, enregistrer une dépense consiste à donner un poste budgétaire, donner une justification du moment de la dépense, et enfin à donner un montant. Concrètement, ces trois étapes ont été séparées, pour que l'utilisateur se préoccupe d'une à la fois. Et dans le cas où le dialogue est celui par manipulation directe (avec les icônes des billets et des pièces), l'étape où l'utilisateur donne le montant est elle-même décomposée au maximum, puisqu'il peut relâcher sa concentration après chaque billet ou pièce introduit. Donc, par ces découpes, il y a peu d'efforts prolongés pour l'utilisateur.

5) Une erreur ne doit jamais entraîner de résultats catastrophiques, et l'utilisateur doit pouvoir corriger facilement.

Après chaque donnée introduite par l'utilisateur, il existe la possibilité de corriger une éventuelle erreur. Soit cette possibilité est explicitement donnée par le programme, qui demande confirmation (d'un montant, d'un poste budgétaire, etc), soit l'utilisateur peut lui-même demander une correction (par exemple, en cliquant sur un bouton "EFFACE").

6) Permettre le retour en arrière, l'utilisateur doit pouvoir recommencer ou abandonner une partie de son travail.

Pour chaque fonctionnalité, l'utilisateur peut, à tout moment, quitter la partie dans laquelle il se trouve, et par conséquent tout ce qu'il peut avoir fait dans cette partie est ignoré, abandonné.

7) Offrir à l'utilisateur le sentiment de contrôler l'application, il doit diriger l'application au maximum.

Cette règle est mise en pratique, par le fait que l'utilisateur, une fois arrivé au menu, peut se "déplacer" librement ; choisir les fonctions qu'il veut, dans l'ordre qu'il veut et peut toujours abandonner la fonction en cours, comme nous l'avons vu au point précédent. Toutefois, au début de toute session, l'utilisateur est davantage guidé, puisqu'il doit nécessairement commencer par donner son nom, puis la somme d'argent dont il dispose (ceci afin de savoir quels "comptes" seront manipulés et pouvoir dire à quel moment les comptes seront le reflet exact de la réalité).

8) Réduire la charge de mémorisation.

Nous avons vu, dans la règle 4, que l'effort de concentration est réduit, et il n'y a pas à proprement parler d'effort de mémorisation, pas de numéro de code puisque l'utilisateur s'identifie en tapant son prénom. Bien entendu le but est d'amener les utilisateurs à enregistrer toutes leurs dépenses et leurs recettes, et cela implique de la mémorisation, mais comme c'est le but du programme, il ne s'agit pas d'une charge de mémorisation inutile, qu'il faut chercher à réduire.

#### 4.4.2. Respect des principes de Hansen

Ceux-ci sont plus généraux, voyons rapidement ce qu'il en est :

\* Tout d'abord il est conseillé de "connaître les utilisateurs" ; comme nous l'avons vu dans le chapitre 3, la démarche que nous avons suivie pour élaborer les spécifications s'efforçait au maximum de connaître les personnes à qui le programme allait être destiné.

\* Le principe suivant "minimiser la mémorisation", préconise notamment de

- permettre la sélection plutôt que l'entrée de données ; c'est le cas dans le menu qui apparaît, puisqu'il suffit de sélectionner avec la souris l'option choisie, sans rien taper au clavier
- utiliser des noms plutôt que des nombres ; or c'est bien par son

prénom que l'utilisateur s'identifie.

\* "optimiser les opérations", il est dit notamment de modifier le moins possible l'affichage pendant l'exécution d'une opération. Comme nous l'avons souligné au chapitre 1, ceci n'exclut pas les feed-back, pour donner des renseignements sur l'effet produit par une action. Donc, pendant une opération, l'affichage ne change pas, et après les feed-back arrivent.

\* le dernier principe "engineer for errors" conseille notamment de

- concevoir l'application de manière à empêcher les erreurs les plus communes ; par exemple, quand une somme est demandée, toute suite de caractères illégale est directement effacée (sans ralentir par un message d'erreur ...etc)
- permettre des actions réversibles ; nous avons vu que quand l'utilisateur est entré dans une fonction, il peut toujours faire machine-arrière, et l'abandonner pour revenir à l'état précédant le choix de cette fonction.
- offrir de la redondance ; les multiples feed-back sont un bon exemple de redondance.



## Chapitre 5. L'architecture et le principe de séparation modulaire

### Introduction

Nous avons présenté le domaine de l'informatique et du handicap, et celui de l'interface homme-machine. Nous avons ensuite décrit le contexte dans lequel s'est déroulé ce travail et la démarche que nous avons suivie pour élaborer les spécifications de l'application. Ensuite, nous avons vu ces spécifications.

Maintenant nous allons voir, sur base de ces spécifications et sur base du principe de séparation modulaire entre les fonctionnalités et l'interface, à quelle architecture nous sommes arrivés. Nous présenterons brièvement les modules qui la composent et enfin nous verrons comment nous avons implémenté cela et quelles possibilités supplémentaires offrait le langage que nous avons utilisé.

### 5.1. Respecter le principe de séparation modulaire

Au chapitre 1, nous avons vu que pour réaliser plusieurs interfaces pour les mêmes fonctionnalités, il y avait lieu de séparer les modules prenant en compte les fonctionnalités de ceux réalisant l'interface. On permet ainsi l'indépendance de conception de l'interface par rapport à la conception des fonctionnalités (qui peuvent être implémentées sans vraiment connaître le matériel ou les techniques d'interaction qui seront utilisées).

Donc c'est à ce niveau que ce principe doit être appliqué. Concrètement, le but est de structurer l'application qui doit être développée, en divers composants (modules) qui ont des relations bien définies. Et parmi ces composants, ceux qui interagissent directement avec l'environnement extérieur (pour saisir ou afficher de l'information), devront être séparés de ceux qui n'interagissent pas directement avec l'extérieur.

On peut exprimer le principe de séparation modulaire autrement, en disant que tous les besoins propres aux dialogues doivent être regroupés dans des modules séparés de ceux qui réalisent les besoins propres aux fonctionnalités de l'application.

## **5.2. L'architecture logique**

Le but est donc de structurer l'application de manière hiérarchique pour une relation donnée. Notre hiérarchie se base sur la relation UTILISE, qui dit qu'un composant A utilise un composant B si et seulement si la validité de A dépend de la disponibilité d'une version correcte de B. Rappelons également que la relation "appelle" n'est valable que du point de vue de l'implémentation, c'est-à-dire de l'architecture physique.

Voyons concrètement l'architecture logique que nous avons élaborée, en rappelant que, comme dans toute application, il n'y a pas une et une seule solution et que celle-ci est une architecture possible parmi d'autres.

### **5.2.1. Hiérarchisation et découpe en modules**

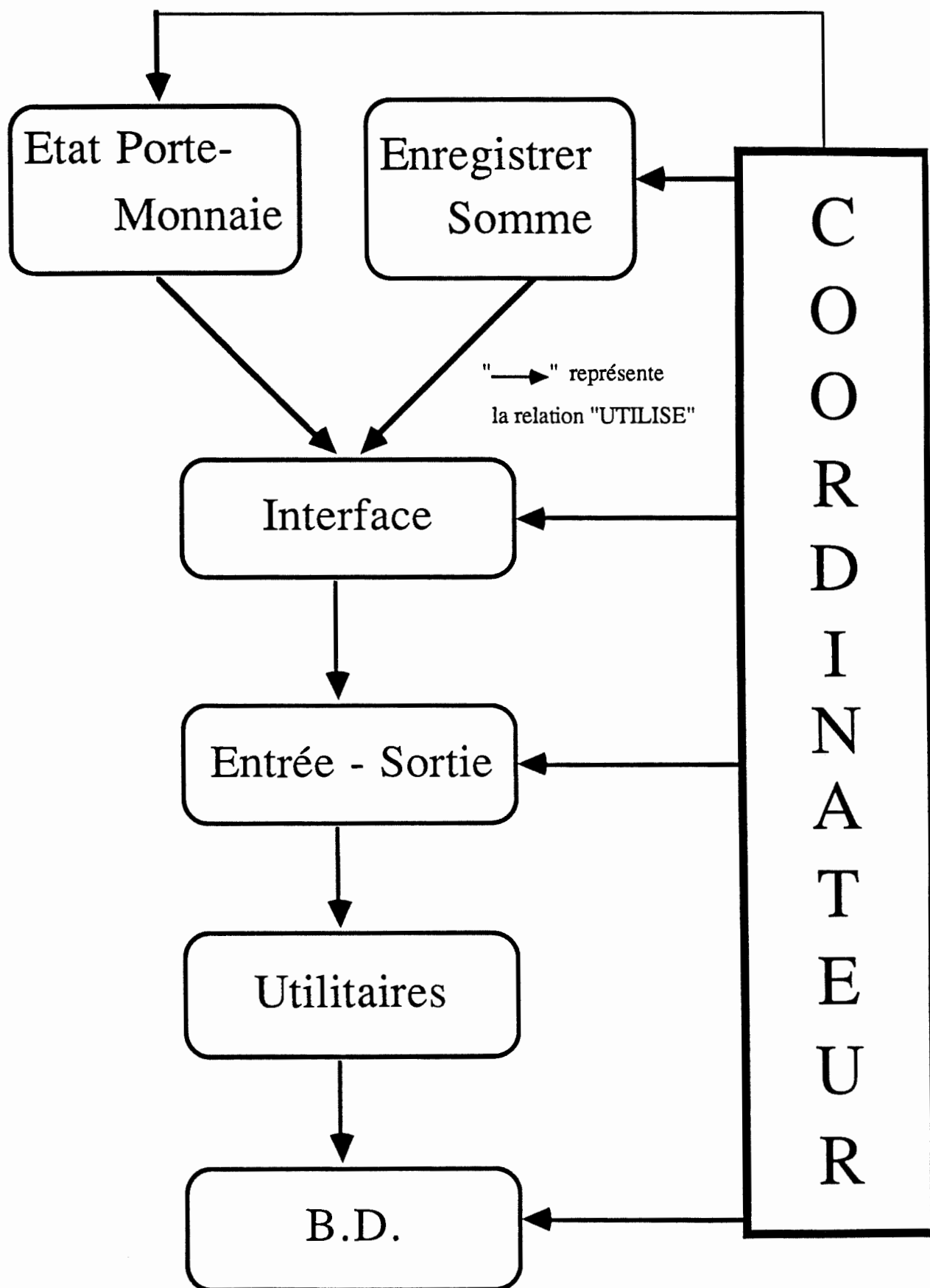
Dans cette section, nous présenterons le graphe représentant l'architecture logique (selon la relation "utilise"), en donnant une description succincte de chaque module. Ils seront mieux spécifiés à la section suivante.

Comme le suggère la présentation graphique, on peut relever des niveaux dans l'architecture.

Tout d'abord, celui que l'on peut appeler le niveau des modules fonctionnels. A ce niveau on trouve 2 modules :

- \* "Etat-porte-monnaie" correspond à la fonction de l'application qui permet de visualiser le contenu du porte-monnaie et de rappeler la différence qu'il y a entre cette valeur "réelle" et le solde actuel des comptes. Ce module correspond donc exactement à une fonctionnalité de l'application.

- \* "Enregistrer-somme", ce module par contre réalise une recombinaison de trois fonctionnalités de l'application ; celles d'enregistrement des dépenses, d'enregistrement des recettes et de la somme (initiale) dont l'utilisateur dispose réellement au début de sa session de travail.



Au niveau suivant, nous avons un seul module, que nous avons appelé "Interface", et qui est un coordinateur du dialogue. En fait, c'est lui qui permet de réaliser le principe de séparation modulaire entre l'interface et les fonctionnalités. Les modules du niveau fonctionnel prennent en charge les fonctionnalités, mais ils ne "connaissent" pas le type de dialogue, les éléments qui ont été combinés par l'éducateur.

Leur rôle est, par exemple, d'enregistrer une recette ; ce qui implique dans ce cas, de l'ajouter au solde actuel des comptes, mais le dialogue mis en oeuvre pour interagir avec l'utilisateur en vue de saisir cette somme n'est pas du ressort de ces modules. Ceci veut dire qu'il sera possible d'implémenter ces modules sans connaître les techniques d'interaction qui seront utilisées, et que bien entendu, plusieurs dialogues pourront être "associés" à ces fonctionnalités.

Donc la justification de cette découpe est à trouver dans le principe de séparation modulaire entre interface et fonctionnalités. Concrètement, le module Interface regardera quels paramètres ont été choisis par l'éducateur et utilisera les services offerts par les modules des niveaux inférieurs, pour réaliser effectivement le dialogue. C'est donc lui qui coordonnera le dialogue associé aux fonctionnalités, en utilisant les services offerts par les autres modules.

Au niveau inférieur, nous avons un module d'"Entrée-Sortie", qui offre un éventail de services, pour réaliser concrètement un dialogue ou l'autre. L'avantage est que si de nouveaux besoins sont exprimés au niveau du dialogue, il suffit d'ajouter de nouveaux services à ceux déjà offerts. Il s'agit réellement d'un catalogue de services. Et bien entendu, la manière d'implémenter ce module est invisible pour les modules de niveau supérieur, ils doivent juste connaître les spécifications, c'est-à-dire dans ce cas les services offerts et la façon de les utiliser.

Remarquons que rien n'empêche, a-priori, que les modules du niveau fonctionnel utilisent ceux de ce niveau (ce serait pour le cas où un seul dialogue est proposé pour une fonctionnalité, dès lors l'"intermédiaire" du module Interface serait relativement inutile). En fait, nous verrons que c'est à peu près ce qui se passe quand le Coordinateur (qui sera présenté ci-dessous) utilise directement un service offert par ce module, pour saisir le prénom de l'utilisateur.

Au niveau inférieur, nous avons regroupé un certain nombre de

services de base relatifs au dialogue et qui peuvent être utilisés par plusieurs dialogues effectifs, comme par exemple la gestion du thermomètre de couleur. A ce niveau, il s'agit réellement d'utilitaires, qui comme leurs noms l'indiquent, seront largement utilisés par les modules de niveau supérieur.

Enfin, nous avons le module qui centralise les données qui doivent être mémorisées, mais il ne s'agit pas d'une grosse base de données comme on l'entend habituellement.

Il faut remarquer qu'à ce moment nous avons une découpe qui ne fixe pas du tout le scénario d'exploitation de l'application ; rien ne dit quelle séquence sera suivie pendant une session de travail. A-priori, cette architecture peut supporter plusieurs scénarios d'exploitation différents. Un scénario d'exploitation particulier sera réalisé, sera pris en charge par ce que nous appelons le Coordinateur. C'est ce module qui réalisera donc l'algorithme que nous avons présenté à la section 4.2.2., et qui a été choisi comme base de toute session de travail. Si plus tard un autre scénario est préféré, les changements seront concentrés dans le Coordinateur et dans ce module uniquement.

### 5.2.2. Spécifications externes des modules

Voyons maintenant un peu plus précisément (sans toutefois entrer dans les détails) les spécifications des différents modules identifiés dans l'architecture logique. Pour plus de compréhension, nous avons présenté ces spécifications sous la forme suivante : arguments reçus par le module, préconditions portant sur ces arguments, résultats, postconditions sur ces résultats. Mais tout en exprimant ces "assertions" en langue française.

#### \* Le module "Enregistrer-Somme"

Comme nous l'avons vu, il réalise une recomposition de trois fonctionnalités assez semblables de l'application. Voyons ce qu'il reçoit en argument et ce qu'il donne en résultat :

Arguments : la fonctionnalité à réaliser, la base de données (BD)

Préconditions : la fonctionnalité sera soit "enregistrer une recette",

"enregistrer une dépense" ou "enregistrer la somme initiale".

Résultats : Etat de sortie, Somme, BD (mise à jour)

Postconditions : si Etat de sortie est faux, aucune somme n'est saisie (cela signifie que l'exécution du module a été interrompue par l'utilisateur), la BD reste inchangée, si Etat de sortie est vrai, un réel a été enregistré dans Somme, et la BD est mise à jour suivant la fonctionnalité à réaliser, qui avait été reçue en argument (donc soit une somme initiale est sauvée dans la BD, soit le solde des comptes est mis à jour et l'enregistrement comptable (somme, moment, poste) est sauvé).

#### \* Le module "Etat-Porte-Monnaie"

Arguments : la BD.

Résultat : la somme initiale a été présentée à l'utilisateur, de même que la différence avec le solde actuel des comptes, la BD.

Postconditions : la BD reste inchangée.

#### \* Le module "Interface"

Arguments : la liste des paramètres relatifs à l'utilisateur, une fonction à réaliser.

Préconditions : la fonction à réaliser est soit "saisie du nom", "menu", "recette", "dépense", "somme initiale", "moment",

"poste dépense", "poste recette", "etat porte-monnaie" (+ les cas où il y a une différence entre la réalité et les comptes), "récapitulation des recettes et dépenses".

La liste des paramètres est soit celle fixée par un éducateur, soit celle par défaut.

Résultats : réalise la fonction demandée via le dialogue choisi

### \* Le module "Entrée-Sortie"

Arguments : une fonction et le mode de dialogue associé

Préconditions : la fonction doit être une des suivantes : "saisie du nom", "saisie d'une somme avec monnaie suisse" (manipulation directe), "saisie d'une somme avec monnaie belge", "saisie d'une somme par le clavier", "visualisation de l'état du porte-monnaie par nombre", "par dessins (monnaie suisse)", "par dessins (monnaie belge)", "saisie d'une poste budgétaire de recette", "saisie d'un poste budgétaire de dépense", "saisie d'un moment du jour", "saisie d'un jour de la semaine", "menu", "récapitulatif des recettes et des dépenses enregistrées" et "équilibre".

Remarque : il est donc facile d'étendre la liste des services offerts par ce module, et facile aussi de localiser quel service réalise quelle fonction (ceci dans un souci de facilité de maintenance).

Résultats : réalisation du service demandé.

### \* Le module "Utilitaires"

Arguments : l'utilitaire désiré, la BD

Préconditions : la liste actuelle des utilitaires est la suivante : "thermomètre de couleur", "règle du temps d'une semaine", "règle du temps d'un jour".

Résultats : réalisation du service demandé en argument, BD inchangée.

## 5.3. A propos de l'architecture physique

Dans cette section nous allons voir rapidement quels outils ont été choisis pour développer physiquement l'application et quelques choix relatifs à l'architecture physique qui implémente celle logique.

### **5.3.1. Modula-2**

Il a été très vite fixé, avec l'aide de spécialistes de l' Amiga, que l'environnement de programmation (pour Amiga) le plus adapté à ce que nous voulions réaliser, était le Modula-2 proposé par Benchmark. Voyons rapidement les concepts de base de ce langage et ses avantages évidents.

Modula-2 est le troisième langage créé par Niklaus Wirth (le premier est le Pascal). Comme son nom l'indique, il est basé sur le concept de "module". Un module sera un groupe de procédures et de structures de données, dont certaines seront accessibles (offertes) à d'autres modules.

Sans revenir en détail sur les avantages des modules, disons qu'ils permettent une structuration, une hiérarchisation de l'application qui la rendra plus facile à développer et à maintenir. Les modules pourront être développés (et dans ce cas, compilés) séparément. La lisibilité, la traçabilité du programme est accrue, il est en effet plus facile de voir dans quelle(s) partie(s) est implémentée telle opération, telle module logique (ceci dans un but de maintenance). De plus, les procédures qui sont dépendantes du matériel utilisé ("machine-dependent") pourront être regroupées, isolées dans des modules particuliers, pour faciliter la portabilité de l'application ; en effet, si le programme est "porté" sur une autre machine, seuls ces modules devront être réécrits. Ce sont quelques exemples des avantages de la modularisation, que permet le Modula-2.

#### **5.3.1.1. Particularités du Modula-2**

Syntaxiquement, il est très proche du Pascal, ce qui signifie notamment une assez bonne lisibilité du code source pour toute personne amenée à maintenir le programme. Cette lisibilité est augmentée par la modularisation.

Ensuite, il faut savoir qu'un module est composé de 2 parties séparément compilables : un "definition module" et un "implementation module" correspondant. Le premier spécifie les objets (procédures et structures de données) exportés, offerts par le module, le deuxième contient les codes et les données qui implémentent les objets spécifiés dans le "definition module". En somme, le "definition module" spécifie ce que fait le module et l'"implementation module" spécifie comment le module le fait.



Si nous nous attardons sur ce point, c'est que les retombées de cette séparation sont importantes, en effet :

- \* un "definition module" spécifie ce que fait le module mais cache les détails d'implémentation. Il peut être utilisé comme documentation du module.
- \* un "implementation module" peut être modifié ou remplacé sans devoir recompiler aucun autre module dans le système, parce que le "definition module" ne change pas. Ceci est très agréable pour le développement et la maintenance.
- \* et surtout, les "definition modules" peuvent être écrits et compilés avant que les "implementation modules" ne soient écrits. C'est très intéressant pour un projet avec plusieurs personnes qui se répartissent la programmation, parce que les "definition modules" fixent très vite les interfaces entre les modules. Et quand ces interfaces sont fixées, tout le monde doit s'y tenir (or on sait combien les interfaces entre modules peuvent poser de problèmes quand elles ne sont pas clairement définies avant de se répartir la tâche).

Par conséquent, nous avons pu très vite fixer l'architecture physique implémentant l'architecture logique décrite avant, et en tirer les "definition modules" correspondants.

### 5.3.2. Choix d'une architecture physique

Voyons concrètement les choix que nous avons faits pour implémenter chaque module identifié dans l'architecture logique. Dans les annexes, nous avons repris les "definition modules" des modules physiques.

- Le module Coordinateur sera donc le corps principal du programme.
- Au module "Etat-Porte-Monnaie" correspondra le module physique du même nom et comme on peut le voir dans son "definition module", il offre une procédure (avec les paramètres nécessaires).
- Le module correspondant au module logique "Enregistrer-Somme", porte lui aussi le même nom. Nous avons vu que celui-ci est la recombinaison de 3 fonctions de l'application, donc très logiquement, le module physique offre 3 procédures réalisant ces trois fonctions.

- Le module Interface rappelons-le est celui qui regarde les paramètres relatifs au dialogue homme-machine et qui utilisera la ou les procédures adéquates des modules de niveau inférieur. Le module physique offre 12 procédures qui font ce travail pour 12 fonctions ou parties de fonctions de l'application (celles relevées lors de la spécification du module logique Interface, à la section 5.2.2.). Donc dans chaque procédure, il s'agira de voir le dialogue choisi et concrètement d'appeler les services correspondants, offerts par les modules de niveau inférieur.

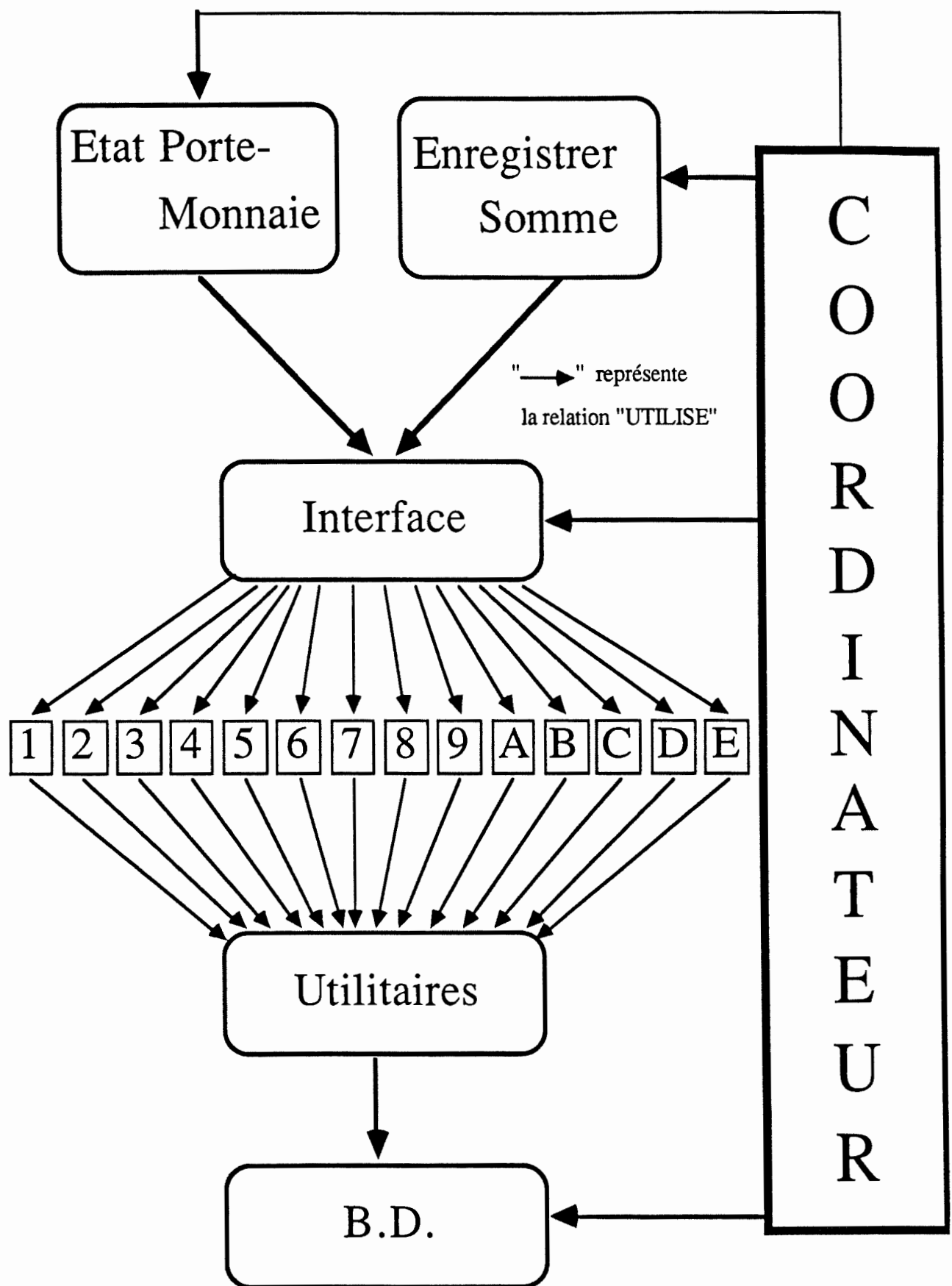
- Jusqu'ici la correspondance entre modules logiques et modules physiques est très "naturelle" ; pour le module "Entrée-Sortie", il en va autrement, mais nous allons voir que c'est dû à des raisons purement pratiques. Vu la liste de services identifiés dans les spécifications de ce module logique, il paraîtrait évident d'en faire un module physique offrant autant de procédures que de services identifiés. Et c'est ce que nous avons choisi dans un premier temps.

Malheureusement pour la lisibilité, il s'est très vite avéré qu'en pratique cela compliquait beaucoup les choses. L'"implementation module" est vite devenu trop grand et des lenteurs d'édition, de programmation et de compilation se sont faites jour. Pour ces raisons pratiques, nous avons décidé d'isoler chaque service (spécifié dans le module logique) dans un module physique, qui offre donc une seule procédure. Par conséquent, au module logique "Entrée-Sortie" correspondent 14 modules physiques.

- Par contre pour le module "Utilitaires", la correspondance physique, est un module qui offre autant de procédures que de services identifiés dans les spécifications.

- Finalement la BD très simple, est réalisée par le module "Variables-Globales" accessible par tous les modules.

Ayant écrit les "definition modules", nous avons fixé les interfaces et il était possible de nous répartir la tâche de programmation.



- 1 : Récapitulatif Recettes - Dépenses
- 2 : Saisie du nom
- 3 : Saisie monnaie suisse
- 4 : Saisie monnaie belge
- 5 : Saisie par le clavier
- 6 : Porte-monnaie par nombre
- 7 : Porte-monnaie suisse
- 8 : Porte-monnaie belge
- 9 : Poste budgétaire dépense
- A : Poste budgétaire recette
- B : Moment de la journée
- C : Jour de la semaine
- D : Menu
- E : Equilibre

### **5.3.3. Remarques sur l'environnement de programmation**

Avant de terminer ce chapitre, nous voudrions parler d'une possibilité que nous donne l'environnement de programmation Modula-2 que nous avons utilisé. Cette facilité a trait à la façon d'intégrer les dessins dans un programme. En effet, il est possible d'écrire le programme, de le compiler sans avoir dessiné les parties graphiques qui apparaîtront. Celles-ci peuvent être dessinées et ensuite "associées" au programme compilé.

Cette façon de procéder est très intéressante, puisque si un dessin doit être modifié (par exemple parce qu'il n'est pas compris par les utilisateurs), on le modifie avec un logiciel de dessin, et on le ré-associe au programme, sans que celui-ci doive être recompilé.

Donc, il s'agit aussi d'une espèce de séparation entre l'interface graphique et le programme informatique. De plus, les logiciels de dessin classiques peuvent être utilisés par les utilisateurs eux-mêmes. Ce ne fut pas le cas dans notre projet, mais on peut donc imaginer que les éducateurs et les personnes handicapées se mettent ensemble et (re)dessinent les parties graphiques, de la façon qui leur convient le mieux. Il sera donc facile (sans recompiler) d'intégrer ces dessins dans le programme.

## Chapitre 6. La phase d'évaluation

### Introduction

Le chapitre 3 nous a montré comment parvenir à de bonnes spécifications en faisant participer psychologues et utilisateurs finals dans un processus essentiellement itératif. Une fois les spécifications élaborées (et présentées en détail dans le chapitre 4), nous sommes logiquement passés à la phase de développement, d'implémentation qui a été exposée dans le chapitre 5.

La phase suivante est celle de l'évaluation du programme réalisé, pour voir notamment, dans quelle mesure il répond aux objectifs que nous avons définis avant le lancement du projet, à savoir, que le logiciel constitue un outil d'autonomie et un outil d'apprentissage.

Ce chapitre 6 nous montrera d'abord où en sont aujourd'hui les recherches en matière d'évaluation de logiciels et en particulier de logiciels pour handicapés mentaux. Nous verrons ensuite comment, dans notre projet, nous avons mis sur pied cette phase d'évaluation que nous avons scindée en deux parties (l'évaluation de l'interface et l'évaluation pédagogique) et quels sont les deux outils dont nous avons fait usage (les traces et les questionnaires). La quatrième section nous expliquera comment s'est déroulée l'expérimentation et comment nous avons exploité les résultats obtenus.

### 6.1. L'état de l'art en matière d'évaluation de logiciels

Nos recherches concernant l'évaluation de logiciels ne nous ont donné que peu de résultats. Nous ne dirons pas qu'il s'agit d'un domaine nouveau (car on trouve des travaux datant de 1976), mais bien qu'apparemment peu de chercheurs s'aventurent dans ce domaine. Il faut néanmoins souligner les travaux menés par les Drs Lynn Lyons Morris et Carol Taylor Fitz-Gibbon du Centre de Recherche sur l'Evaluation de l'Université de Californie à Los Angeles qui ont débouché sur un kit de sept ouvrages ayant pour unique sujet l'évaluation de programmes sous tous ses angles :

- \* Evaluator's Handbook
- \* How to focus an evaluation

- \* How to design a program evaluation
- \* How to use qualitative methods in evaluation
- \* How to assess program evaluation
- \* How to measure attitudes
- \* How to measure performance and use tests
- \* How to analyse data
- \* How to communicate evaluation findings

Il faut noter que ce kit n'a pas seulement trait à l'évaluation de programmes informatiques, mais de toutes sortes d'actions telles une méthode de cours, une action ayant pour but de changer les attitudes des consommateurs, etc. Nous pensons néanmoins que ces travaux peuvent constituer une base intéressante pour notre réflexion.

L'ouvrage n° 3 du kit nous permet d'examiner ce que les auteurs préconisent comme façon de procéder. Il y expliquent comment élaborer un design d'une évaluation, celui-ci étant défini comme un plan spécifiant qui doit être mesuré et quand, c'est-à-dire quels groupes et à quels moments. Pour évaluer un programme, on constituera deux groupes : l'*experimental group* et le *control group*. L'*experimental group* ou E-group est l'ensemble des entités (étudiants, classes, écoles, employés, sections,...) qui reçoivent le programme à évaluer. Le *control group* ou C-group est défini comme l'ensemble des entités qui sont aussi similaires que possibles aux entités du E-group, mais qui ne reçoivent pas le programme à évaluer.

Cette définition du C-group nécessite deux remarques :

- 1) Le C-group doit être aussi similaire que possible au E-group.

Deux moyens sont préconisés pour obtenir cette similarité :

La première possibilité et la meilleure consiste à former le C-group par assignation random, et l'on obtient alors ce que les auteurs appellent un *true control group*.

Si la randomization n'est pas utilisable, on formera ce que les auteurs appellent un *non-equivalent control group*. Il s'agira de former un C-group en faisant attention d'utiliser la même procédure qui a permis de former le E-group et il sera nécessaire de bien documenter les différences existant entre les deux groupes.

2) On voit dans la définition du C-group que celui-ci ne reçoit pas le programme à évaluer. Que reçoit-il donc ? Il existe plusieurs possibilités que les auteurs classent par ordre décroissant de qualité, c'est-à-dire que plus on se trouve haut dans cette liste, plus notre design sera puissant :

- un programme qui est concurrent du programme à évaluer  
c'est-à-dire qui a les mêmes buts et qui utilise les mêmes moyens
- un programme qui a les mêmes buts que le programme à évaluer
- pas de programme du tout.

Ayant ces deux groupes, on utilisera des tests, questionnaires, observations que l'on administrera à ceux-ci. On pourra alors comparer les résultats des tests afin d'évaluer le programme. On peut se demander quand il est utile de tester ces deux groupes.

Tout d'abord, certains tests ou mesures peuvent être collectés avant que les participants reçoivent le programme ou commencent l'expérience. C'est ce que les auteurs appellent les *pretests* . On utilisera les *pretests* pour servir différents buts :

- \* pour vérifier ou assurer la similarité des groupes
- \* pour fournir une base pour vérifier les gains faits pendant un programme

Un deuxième type de tests sont les *posttests* qui sont des mesures prises à la fin de l'expérience. C'est dans les scores du *posttest* que les résultats de l'évaluation vont apparaître. Le *posttest* est la variable dépendante ; les résultats du *posttest* dépendent, partiellement au moins, de ce qui s'est passé pendant le programme.

On peut encore mentionner les *time series tests* qui sont des séries de tests donnés à intervalles égaux avant et après le programme.

Selon que l'on ait opéré une assignation random ou non, que l'on ait donné des *pretests*, des *posttests* et/ou des *time-series tests*, on trouvera des designs différents :

QUI EST MESURE ?				
QUELLES MESURES SONT FAITES ?		<u>E-group seulement</u> Ces designs peuvent seulement répondre à des questions qui ont trait à la façon dont un programme marche.	<u>Plus d'un groupe</u> Ces designs peuvent répondre à des questions comparant les effets du programme avec une alternative.	
			True control group	non-equivalent control group
	PRETEST ET POSTTEST	DESIGN 6	DESIGN 1	DESIGN 3
	SEULEMENT POSTTEST	pas recommandé	DESIGN 2	pas recommandé
	TIME-SERIES TESTS	DESIGN 4	bon, mais inhabituel	DESIGN 5

Nous nous arrêtons ici en ce qui concerne l'évaluation proposée par le Centre de Recherche sur l'Evaluation de l'Université de Californie. Le but se limite à voir quelle est la façon de procéder. Nous verrons par la suite en quoi notre démarche s'est approchée d'un de ces designs.

## **6.2. Les deux types d'évaluation préconisés**

On peut se demander ce dont il s'agit lorsqu'on parle d'évaluation de logiciels interactifs pour handicapés mentaux. Il y a, selon nous, deux façons de voir les choses. on peut se placer d'une part du côté des concepteurs et d'autre part du côté des psychologues.



### 6.2.1. Le point de vue interface

Si l'on choisit une vision conceptuelle, on se demandera davantage si le dialogue et les fonctionnalités ont été correctement choisis. Nous ne parlons pas ici de la correction du programme par rapport aux spécifications. Nous posons l'hypothèse que le programme réalise correctement les fonctionnalités. Il s'agit plutôt de savoir si la logique de déroulement du programme est correctement perçue et si les dialogues tels qu'ils sont proposés sont compris par l'utilisateur handicapé. On pourra ainsi chercher à savoir si telle erreur est souvent commise par un grand nombre d'utilisateurs, si un pictogramme ne représente pas correctement la notion qui y est associée, si le fait de laisser libre le déroulement du programme (par des menus, plutôt qu'un déroulement imposé où la séquence des opérations est préfixée) ne perturbe pas les utilisateurs, etc. Ainsi, par exemple, si l'on remarque qu'il arrive assez fréquemment que les utilisateurs sélectionnent par erreur l'option FIN dans le menu graphique, on peut en déduire que quelque chose ne va pas à ce niveau. Evidemment, l'utilisateur ne pouvant expliquer lui-même pourquoi il s'est trompé, on ne trouvera pas nécessairement la solution idéale au premier essai. On pourra essayer divers moyens, tels que changer le pictogramme associé à cette fonction, éloigner le pictogramme FIN des autres options du menu, mettre en évidence les pictogrammes des autres fonctions en les entourant d'un cadre, demander une confirmation lorsque l'option FIN est sélectionnée,... Il faut noter que, dans notre cas particulier, les fonctionnalités étant assez réduites, l'évaluation dans une optique conceptuelle a davantage eu trait au dialogue et c'est pourquoi nous l'avons appelée "Evaluation de l'Interface".

### 6.2.2. Le point de vue pédagogique

Les psychologues vont davantage s'intéresser à l'utilité du programme. Ils vont dès lors se demander si le programme atteint les buts que l'on s'était fixés quand la décision de lancer le projet a été prise. Le but est donc pour eux de mesurer les gains obtenus par les utilisateurs grâce à l'utilisation de ce programme. Ont-ils une meilleure compréhension des notions de recette, dépense, tenue des comptes ? Certains

deviennent-ils autonomes en ce qui concerne la tenue des comptes ? Est-ce que le programme a permis des évolutions aux points de vue affectif, intellectuel, culturel,... ? Il est évident qu'il est difficile pour les informaticiens de se poser les bonnes questions à ce sujet. Il s'agira principalement d'un travail laissé aux psychologues comme nous l'expliquerons dans la section suivante. Les résultats de ce type d'évaluation sont très importants car ils peuvent entraîner jusqu'au rejet pur et simple du programme réalisé si l'on arrive à la conclusion que les handicapés apprennent davantage de choses sans l'outil informatique. Nous avons appelé ce type d'évaluation "l'évaluation pédagogique".

Ces deux types d'évaluation peuvent donner lieu à des modifications des spécifications (des fonctionnalités ou de l'interface) ou à de nouvelles spécifications. Les résultats de ces évaluations permettent donc de commencer une nouvelle itération du cycle de vie du projet : développement des spécifications, implémentation de celles-ci, évaluation du nouveau produit obtenu (nous avons en effet souligné dans le chapitre 2 que le développement d'un projet informatique pour handicapés mentaux est essentiellement un processus itératif).

### 6.3. Les moyens à notre disposition

Nous allons voir maintenant les moyens que nous avons utilisés pour mener la phase d'évaluation du projet. Il s'agira d'une part de garder des traces de toute session de travail et d'autre part de recueillir des informations au moyen de questionnaires remplis par les éducateurs. Mais nous tenons à insister directement sur l'importance que garde le côté informel, les rencontres, les discussions avec les psychologues, les éducateurs et dans la mesure du possible avec les utilisateurs. Les deux moyens que nous verrons en détail, sont une manière relativement formelle et rigide de recueillir de l'information utile à l'évaluation (et la subséquente amélioration) du programme. Ils ne doivent pas tendre à éliminer les rencontres entre personnes, qui d'une manière plus informelle, apportent de nombreuses informations qui seraient difficilement révélées par un questionnaire ou une trace de manipulation. De plus, bien souvent, ces moyens permettent de relever des points noirs qui seront résolus via une discussion avec les éducateurs. Donc il s'agira souvent de déclencheurs.

### 6.3.1. Les traces de manipulations

Voyons maintenant le premier moyen que nous avons mis en oeuvre pour recueillir des informations intéressantes pour l'évaluation du programme. Nous verrons tout d'abord ce que nous entendons par "traces de manipulations" et le genre de données qu'elles contiennent, à qui elles peuvent être utiles et enfin comment il est possible de les exploiter. Nous verrons concrètement les expérimentations qui ont été menées, à la section 6.4.1.

#### 6.3.1.1. Pourquoi des traces de manipulations ?

Comme nous l'avons vu à la section 1.1.3.4. , un programme peut être conçu pour sauver des informations concernant les manipulations de l'utilisateur pendant une session de travail (sans gêner le déroulement de la session). Ces informations peuvent mettre en évidence les causes d'erreurs de manipulation, la séquence des options que l'utilisateur choisit, etc. Nous avons vu que le problème de communication lié aux personnes handicapées empêchait l'utilisateur de formuler explicitement des remarques sur le programme, et que dès lors les critiques sont "implicites", c'est-à-dire qu'à travers des erreurs enregistrées lors d'une session de travail, il sera possible de mettre en lumière des incohérences du programme (au niveau des fonctionnalités ou de l'interface). L'ordinateur offre donc une grande facilité pour garder ces traces et assure une uniformité qui ne serait pas possible si c'étaient les accompagnants qui prenaient en charge le suivi des manipulations (travail lourd et fastidieux, par ailleurs). Donc, à moindre coût, il est possible par ce moyen de récolter des informations qui peuvent être très intéressantes. Voyons ce que nous avons voulu retenir dans ces traces.

#### 6.3.1.2. Que contiennent les traces ?

Concrètement, après chaque session de travail, un fichier texte est

sauvé. Ainsi donc, chaque fois qu'un utilisateur emploie le programme, une nouvelle trace est créée puis sauvée.

Lors d'une session de travail, l'utilisateur effectue une série de manipulations, c'est cette série chronologique que nous avons décidé de garder dans la trace. Bien entendu, le genre de manipulations faites par l'utilisateur dépend fortement du type de dialogue qui a été mis en oeuvre (c'est-à-dire des paramètres que l'éducateur avait choisis ; parfois une recette sera introduite en une manipulation au clavier, parfois cette opération sera décomposée en plusieurs manipulations de la souris).

Par conséquent, la trace donne la liste des paramètres actifs pour la session en question. Voici un exemple de trace qui serait gardée après une session où l'utilisateur donne une somme initiale de 122 frs, puis enregistre une recette de 124 frs et une dépense de 2 frs, enfin clôture sa session. Comme l'indiquent les paramètres, l'utilisateur introduit les recettes par manipulation directe avec la souris (donc en cliquant sur les icônes des billets et pièces affichées à l'écran) et les dépenses à l'aide du clavier.

- Denis : trace n°5
- Il donne le nom : Denis
- Il confirme le nom
  
- Liste des paramètres choisis
- Monnaie belge
- Questions posées par écrit + parole
- Recettes saisies par souris
- Dépenses saisies par clavier
- Porte-Monnaie -> nombres + dessins + parole
- Thermomètre présent
- Notion de temps présente
- Postes budgétaires à spécifier
- Récapitulatif présent
- Cycle d'une semaine
- Début de cycle = mercredi
- Il appuie sur 100 Frs

- Il appuie sur 20 Frs
- Il appuie sur 1 Frs
- Il appuie sur 1 Frs
- Il appuie sur FINI
- Il confirme la demande
- Il donne une somme initiale de : 122 Frs
- Il choisit la fonction : recettes
- Il appuie sur travail
- Il confirme
- Il donne le poste : Travail
- Il appuie sur lundi
- Il confirme
- Il donne le jour : Lundi
- Il appuie sur 100 Frs
- Il appuie sur 50 Frs
- Il demande d'effacer
- Il confirme la demande
- Il appuie sur 20 Frs
- Il appuie sur 1 Frs
- Il appuie sur 1 Frs
- Il appuie sur 1 Frs
- Il appuie sur FINI
- Il infirme la demande
- Il clique sur 1 Frs
- Il appuie sur FINI
- Il confirme la demande
- Il donne une recette de : 124 Frs
- Il choisit l'option : dépenses
- Il appuie sur divers
- Il confirme
- Il donne le poste : Divers
- Il appuie sur lundi
- Il confirme
- Il donne le jour : Lundi
- Il appuie sur FINI
- Il donne une dépense de : 124 Frs

- Il appuie sur FIN
- Il confirme sa demande
- Les comptes correspondent à la réalité

### 6.3.1.3. Quelles informations peuvent-elles fournir ?

Voici quelques exemples d'informations qu'a-priori les traces pourraient donner, en vue d'évaluer et d'améliorer le programme.

1) Nous avons vu que l'utilisateur est libre de choisir les options qu'il veut, dès qu'il a atteint le menu. Il est intéressant de voir comment il travaille, donc les traces permettent de relever la séquence des actions effectuées. Si un déroulement "type" apparaît, il pourrait éventuellement être fixé, pour les utilisateurs qui ont des difficultés à poser un choix.

2) Voir où les erreurs sont le plus fréquemment commises. Comme on le voit dans l'exemple, quand l'utilisateur infirme une demande, ou demande d'effacer la dernière opération, ou de quitter la fonction dans laquelle il se trouve, tout cela est enregistré dans les traces.

Des erreurs fréquentes peuvent indiquer que des concepts ne sont pas bien compris par les utilisateurs. Ceci peut provenir d'une inadéquation entre le dialogue et l'utilisateur. Par exemple s'il arrive souvent que des utilisateurs quittent la fonction d'enregistrement d'une recette sans avoir mené à bien la fonction, c'est peut-être parce que dans le menu, l'option "recette" est mal représentée et prête à confusion. Par conséquent, l'amélioration porterait sur une manière plus adéquate de représenter le concept de recette et de présenter le menu.

Autre exemple, dans la monnaie suisse il y a une pièce de 5 francs et une de 5 centimes. S'il s'avère que plusieurs fois des utilisateurs cliquent sur la pièce de 5 frs, l'effacent, puis enregistrent une pièce de 5 centimes, cela dénoterait d'une confusion entre les deux icônes et donc des efforts devraient être fournis pour les améliorer.

Donc ces erreurs peuvent révéler des inadéquations des dialogues avec les utilisateurs et amener des modifications.

3) On peut aussi déceler si certains éléments de dialogue sont mieux compris que d'autres, c'est-à-dire s'ils engendrent moins d'erreurs. Soit cela voudra dire que les dialogues plus "mauvais" sont à améliorer, soit

qu'ils ne conviennent tout simplement pas aux utilisateurs qui commettent beaucoup d'erreurs en les utilisant. Ceci nous fait remarquer que l'interprétation des traces n'est pas un processus simple et que parfois seuls les psychologues seront aptes à en tirer certaines conclusions. Nous en reparlerons dans la section suivante, et nous verrons aussi que ces traces peuvent être une aide dans leur travail.

Il est clair que les traces ne permettent qu'une première évaluation de l'interface (qui sera complétée par les questionnaires et bien sûr les rencontres avec éducateurs et utilisateurs), il est clair aussi qu'elles ne contiennent qu'un nombre limité d'informations et que l'expérience montrera sans doute qu'elles devront être enrichies (par exemple, nous n'avons gardé aucune information chronométrique, c'est-à-dire relative au temps que mettent les utilisateurs pour réaliser telle ou telle fonction, ou pour poser tel ou tel choix).

#### 6.3.1.4. A qui les traces peuvent-elles être utiles ?

Bien entendu, ce chapitre traite de l'évaluation et dans ce contexte, les traces ont été développées pour nous aider à tirer des enseignements à distance, de l'utilisation du programme, en vue de modifier les points qui engendrent des problèmes. Toutefois ces traces peuvent être aussi utiles et utilisées par les psychologues et les éducateurs.

En fait, ils ont trouvé nécessaire que les traces contiennent le détail des manipulations effectuées, pour en tirer des conclusions du point de vue pédagogique.

Mais les éducateurs avec qui nous avons travaillé voyaient un autre avantage dans ces traces de manipulations. Elles peuvent aider l'éducateur dans son travail quotidien. En effet, il n'est pas toujours disponible pour vérifier chaque action effectuée par l'utilisateur, car il doit souvent s'occuper d'autres personnes, mais surtout parce que peu à peu le but est de laisser les utilisateurs se débrouiller seuls, pour qu'ils acquièrent une certaine autonomie (ou du moins certains d'entre eux).

#### 6.3.1.5. Graphe des états du programme de tenue des comptes

Ce n'est pas tout d'avoir un programme qui permette de garder des

traces des manipulations effectuées par l'utilisateur, encore faut-il avoir une technique pour les exploiter. Comme nous n'avons rien trouvé dans la littérature à ce sujet, nous avons essayé de définir une façon de travailler.

Nous avons d'abord structuré une session de travail en un certain nombre d'états. Qu'entendons-nous par "état" ?

A un état correspond un écran ou une partie d'écran, qui pose un certain nombre de questions à l'utilisateur, et parmi les réponses, se trouve son choix pour le prochain état (un requester est l'exemple typique de partie d'écran qui correspond à un état, et l'écran qui reste en "toile de fond" n'a aucune importance). Ce choix occasionnera donc une transition vers un autre état où le même processus se répètera.

Dès lors, on peut dire que le programme est une application interactive où l'utilisateur est en quelque sorte guidé à chaque phase de sa session de travail par un écran ou une partie d'écran.

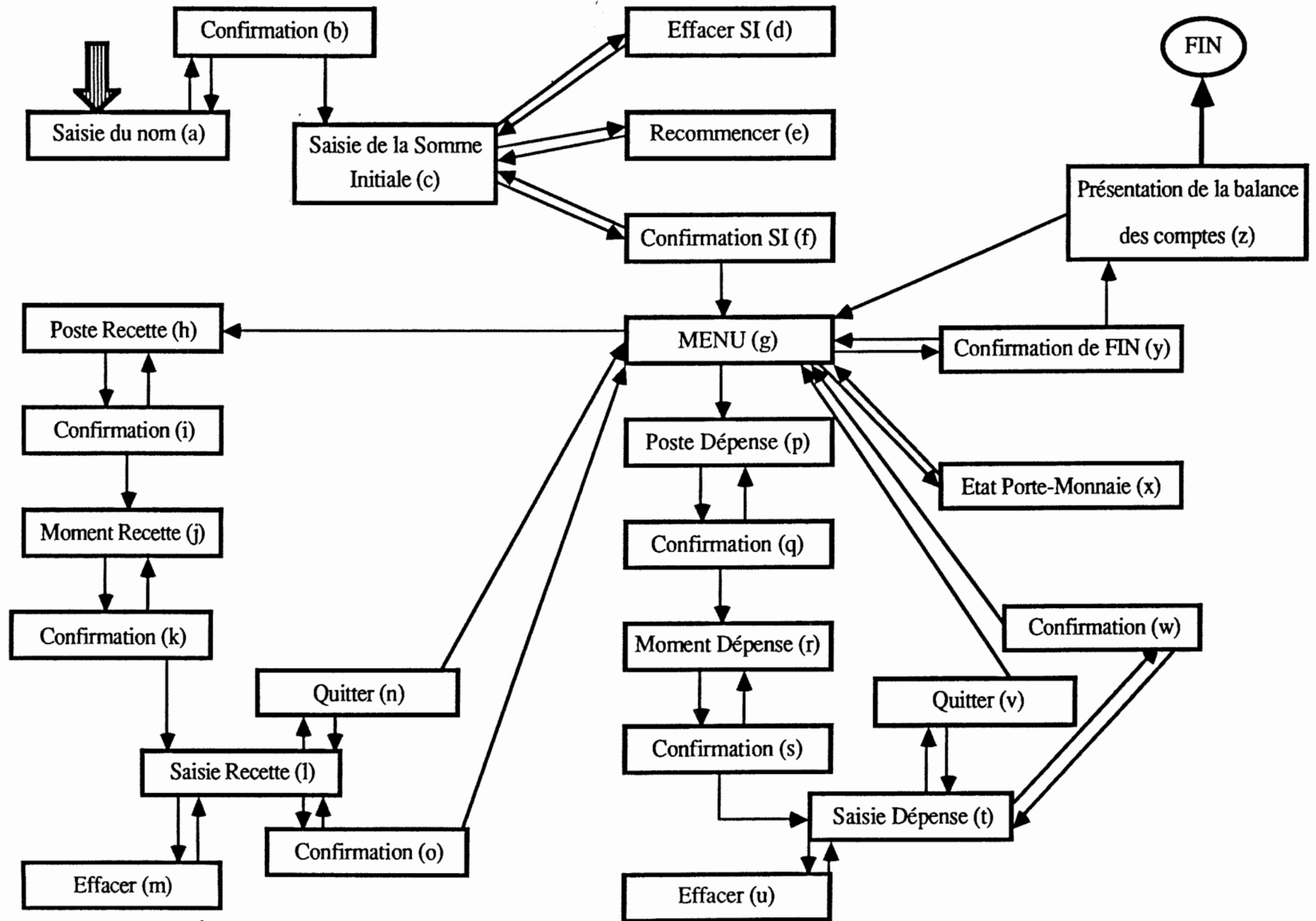
Le processus commence donc par un état initial, passe par un certain nombre d'états intermédiaires, et se termine quand on atteint un des états finals. Dans notre cas, il n'existe qu'un état initial et un état final, mais en toute généralité, pour les applications interactives, il peut y en avoir plusieurs.

Il reste à dire que l'état final, dans notre cas, est un état fictif, et qu'il est absorbant, donc quand on l'a atteint, la session est terminée et par conséquent il n'y a plus moyen d'atteindre les autres états, dans la session de travail actuelle.

Ces notions d'"état" et de "transition" d'un état vers un autre, amènent assez naturellement à représenter la structure d'une session de travail par un graphe de transition. Ce graphe montre tous les états possibles (les noeuds du graphe), ainsi que les transitions entre eux (les arcs reliant certains noeuds).

Bien entendu le graphe d'une session de travail dépend des paramètres choisis, car nous avons vu que , par exemple, si le "temps" n'est pas choisi, l'utilisateur ne doit pas introduire le moment auquel il a fait une dépense ou une recette. Le graphe présenté ci-dessous, est en fait le graphe de transition maximal. De plus pour la facilité d'écriture des sections suivantes, nous avons assigné une lettre de l'alphabet à chaque état identifié. Signalons enfin, qu'on entre dans le processus par la flèche située en haut à gauche dans le graphe.





Nous nous sommes ensuite demandé si nous étions confrontés à un processus Markovien ; pour cela, il fallait que chaque transition ne dépende au maximum que de l'état précédent, c'est-à-dire que la probabilité des transitions d'un état vers un autre, ne dépende pas de l'"historique" des transitions effectuées avant d'arriver à cet état. Donc, il faut que les états par lesquels le processus est passé auparavant, n'aient aucune influence sur les probabilités de transition de cet état vers les autres.

C'est une hypothèse assez forte, qu'il faudrait, sans doute, vérifier expérimentalement. Les données que nous avons ne nous permettaient pas de le faire. Intuitivement, il semble que le processus ne soit pas Markovien ; en effet, lorsqu'on se trouve dans l'état où le menu est affiché, la probabilité de saisir une dépense n'est sans doute pas la même si, auparavant, l'utilisateur a enregistré 20 dépenses ou aucune. Nous ne pouvons donc pas utiliser la théorie des chaînes de Markov.

Il n'empêche, que nous pouvons tout de même étudier les fréquences des différentes transitions entre les états du programme.

#### 6.3.1.6. La matrice des transitions

Nous allons représenter les transitions possibles dans une matrice carrée. Celle présentée ci-dessous se lit horizontalement, c'est-à-dire que la case "cd" (ligne c, colonne d) représente la transition de l'état "c" à l'état "d". Les cases où figure le chiffre 1 désignent des transitions obligatoires (il n'y a pas vraiment de choix). Par contre, là où nous avons mis une croix, cela signifie que l'utilisateur choisira l'état suivant vers lequel il veut transiter, et nous verrons que l'on peut, dans ce cas, essayer d'évaluer les fréquences des différentes transitions. Toutefois, pour faire l'hypothèse que les X ont des valeurs constantes, nous devons supposer que le processus est Markovien, homogène dans le temps. Il s'agit alors de la matrice des probabilités de transition en un pas, du processus étudié.

En tout cas, les cases de la matrice qui contiennent des croix ou des 1, correspondent aux transitions possibles, et celles vides, signifient qu'il n'y a pas de transition possible entre les états correspondant aux coordonnées de la case.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a		1																								
b	X		X																							
c				X	X	X																				
d			1																							
e			1																							
f			X				X																			
g								X								X								X	X	
h									1																	
i								X		X																
j											1															
k									X		X															
l												X	X	X												
m											1															
n							X				X															
o							X				X															
p																1										
q															X		X									
r																		1								
s																	X		X							
t																				X	X	X				
u																				1						
v							X													X						
w							X													X						
x							1																			
y							X																			X
z							X																			

### 6.3.1.7. Les transitions critiques

Voyons parmi les différentes transitions possibles, celles plus ou moins liées à des erreurs et qui peuvent donc être dues à une inadéquation du dialogue. Nous allons les relever en "avançant" dans le graphe, comme si nous étions dans une session de travail (nous désignerons les transitions par les couples de lettres correspondant aux états impliqués, ces couples sont donc les coordonnées des transitions dans la matrice vue ci-dessus).

Au début de la session :

- l'utilisateur commet une erreur dans son nom et le corrige : **ba**
- il demande d'effacer la dernière opération lors de la saisie de la somme initiale : **cd**
- il demande de tout recommencer, lors de la saisie de la somme initiale : **ce**
- il ne confirme pas la somme initiale, après avoir cliqué sur le bouton "FINI" : **fc**

Lors de l'enregistrement d'une recette :

- il ne confirme pas le poste budgétaire qu'il a donné : **ih**
- il ne confirme pas le moment qu'il a donné : **kj**
- il demande d'effacer la dernière opération : **lm**
- il ne confirme pas le montant, après avoir appuyé sur "FINI" : **ol**
- il demande de quitter : **ln**
- il ne quitte pas la fonction après l'avoir demandé : **nl**
- il quitte réellement la fonction "recette" : **ng**

Lors de l'enregistrement d'une dépense :

- il ne confirme pas le poste budgétaire qu'il a donné : **qp**
- il ne confirme pas le moment qu'il a donné : **sr**
- il demande d'effacer la dernière opération : **tu**
- il ne confirme pas le montant, après avoir appuyé sur "FINI" : **wt**
- il demande de quitter : **tv**
- il ne quitte pas la fonction après l'avoir demandé : **vt**
- il quitte réellement la fonction "recette" : **vg**

A la fin du programme :

- il ne confirme pas sa demande de "FIN" de session : **yg**
- il décide de revenir au menu, après présentation de l'écart éventuel entre comptes et réalité : **zg**.

Une autre donnée importante à relever dans la trace : la première option choisie par l'utilisateur quand il arrive au menu. Nous avons vu qu'il est intéressant de voir si une majorité nette se fait jour, marquant ainsi un schéma type qui pourrait être proposé par défaut (pour les utilisateurs qui ont des difficultés à poser des choix).

Nous verrons à la section 6.4.1. les quelques résultats que nous avons

pu tirer des expérimentations menées concernant ces transitions.

### 6.3.2. Les questionnaires

Si Carol Taylor Fitz-Gibbon et Lynn Lyon Morris préconisent l'utilisation de tests pour évaluer les gains obtenus grâce à l'utilisation d'un programme (quel qu'il soit), ceci n'est pas évident quand on se penche sur le problème de l'évaluation de logiciels pour handicapés mentaux. En effet, ceux-ci ne sont pas capables de subir des tests à proprement parler comme un étudiant recevant un programme de mathématique pourrait le faire. Néanmoins, il existe des intermédiaires qui peuvent observer les utilisateurs handicapés et estimer dans une certaine mesure leur degré de compréhension des choses : ce sont les éducateurs. Bien entendu, étant donné qu'il s'agira d'une "appréciation" de ce que les utilisateurs savent et ne savent pas faire, un certain niveau de subjectivité risque d'errer quelque peu les résultats.

L'éducateur va donc indiquer un certain nombre de choses concernant l'utilisateur en répondant à des questions regroupées dans ce que nous appellerons des *questionnaires* ou *grilles d'évaluation*. Si, comme nous l'avons fait, on divise l'évaluation en deux parties, il est normal que deux grilles d'évaluation distinctes soient construites : une grille pour l'évaluation de l'interface et une autre pour l'évaluation pédagogique.

#### 6.3.2.1. L'évaluation pédagogique

##### 1) Construction de la grille

Dans notre projet, nous avons quelque peu participé à la construction de cette grille pédagogique en lançant un premier jet qui a été remodelé par Mesdames Marie-Chantal Wanet et Anne Faton, travaillant au Département de Psychologie (FNDP), en se basant sur deux grilles existantes : une grille en vue d'évaluer des techniques de communication pour handicapés, élaborée par Colette Legrand de l'Université Technologique de Compiègne, et un questionnaire ayant trait à l'évaluation de logiciels (quel que soit leur objet) pour handicapés mentaux, mis sur

pied par André Baechler de la FST et de La Castalie à Monthey.

## 2) Présentation de la grille

La grille est une partie d'un questionnaire proposé en annexe. Il reprend également la grille d'évaluation de l'interface sous une forme quelque peu différente (pp 7 à 15).

## 3) Explication de la grille

Le questionnaire contient, outre la grille d'évaluation de l'interface, quatre autres parties qui sont présentées ci-dessous.

\* Un questionnaire (pp 1 et 2) destiné à l'éducateur qui doit décrire précisément quels sont ses objectifs d'utilisation en ce qui concerne les champs cognitif (connaissances scolaires, culturelles et professionnelles), psycho-moteur (motricité, perception, savoir-faire professionnel) et socio-affectif (affectivité, créativité, autonomie et communication). Ce questionnaire doit être rempli une fois par l'éducateur AVANT le début de l'expérimentation.

\* un autre questionnaire (pp 16 à 19) permet à l'éducateur après l'expérimentation de critiquer le logiciel sur le plan technique et pédagogique, de proposer des améliorations, d'expliquer en quoi ses objectifs de départ ont ou n'ont pas été satisfaits et s'il perçoit de nouveaux objectifs, et enfin de définir les critères d'utilisation du programme c'est-à-dire spécifier quels sont selon lui les prérequis indispensables et à qui il déconseille l'usage de ce programme.

Ces deux parties proviennent du questionnaire élaboré par Monsieur André Baechler.

\* Une autre partie (pp 3 à 6) est appelée la fiche signalétique de l'utilisateur. La première feuille concerne les caractéristiques identifiant l'utilisateur (nom, prénom, âge, sexe, nom de l'institution, statut, type de handicap, Q.I.,...). les pages 4,5, et 6 sont constituées de questions concernant les capacités de l'utilisateur AVANT et APRES l'expérimentation. Ceci permettra donc de déterminer quels sont les gains

réalisés grâce à l'utilisation de ce logiciel. Ces questions couvrent les domaines suivants : les acquis en lecture, en écriture, en calcul, la manipulation de l'argent et la familiarité avec l'ordinateur. C'est cette partie qui constitue l'évaluation pédagogique proprement dite.

\* Enfin, la dernière partie est un questionnaire destiné à l'utilisateur. Bien entendu, ce n'est certainement pas lui qui remplira ce questionnaire mais bien l'éducateur qui essaiera d'obtenir de façon ou d'une autre les réponses à ces questions.

#### 6.3.2.2. L'évaluation de l'interface

##### 1) Construction de la grille.

Par opposition à la première grille, qui était du ressort des psychologues, nous avons pu mettre au point celle-ci nous-mêmes. Nous avons suivi la méthode suivante.

Nous avons tout d'abord établi une grille a-priori, c'est-à-dire que celle-ci contenait des questions concernant des points que nous pensions potentiellement problématiques. Ainsi, par exemple, nous avons expliqué dans les spécifications de l'interface concrète que, lors de la saisie d'une somme par la souris, le fait de cliquer sur un billet a pour effet l'affichage immédiat d'un même billet mais de plus petite taille sur la partie droite de l'écran. Nous avons indiqué que seules cinq occurrences d'un même billet était représentables à l'écran et qu'au-delà de ce chiffre, ce feed-back n'était plus assuré (d'autres continuaient à l'être). On peut évidemment se dire, que cet arrêt soudain du feed-back risque de perturber les utilisateurs. Il était donc normal que la question suivante se trouve dans notre grille a-priori : "La personne n'est-elle pas perturbée que la trace n'apparaît plus lorsqu'elle a entré plus de billets ou pièces que ne le permet l'écran ?".

Une fois cette grille a-priori obtenue, nous nous sommes isolés avec une personne handicapée. L'un de nous commençait une session avec elle et lui expliquait ce qu'elle devait faire et ce qui apparaissait à l'écran. L'autre observait tout ce qui se passait et notait les remarques exprimées par l'utilisateur, son comportement, ses questions, ses erreurs,... Nous avons travaillé de cette façon avec plusieurs handicapés et un certain

nombre de fois avec chacun d'eux. Ceci nous a permis d'obtenir une quantité de rapports que nous avons analysés afin de trouver nombre de questions auxquelles nous n'avions pas pensé lors de l'élaboration de la grille a-priori. Ainsi, par exemple, lors d'un exercice avec une personne handicapée, nous avons pu voir qu'elle a cliqué par erreur sur l'icône représentant l'option FIN, alors qu'elle avait l'intention d'enregistrer une recette. Nous n'avions pas pensé à une erreur possible à ce niveau, ce qui explique qu'aucune question à ce sujet n'apparaissait dans notre grille a-priori. Nous y avons dès lors ajouté la question suivante : "La personne clique-t-elle sur FIN par inadvertance ?"

## 2) Présentation de la grille.

### Questionnaire destiné à l'évaluation de l'interface personne handicapée / ordinateur du programme COMPTES (Version condensée)

Nom de la personne : .....

Sexe : M F (\*)

Age réel : < 4 4-6 6-12 12-16 16-20 > 20 ans (\*)

Niveau de réalisation : < 2 2-4 4-6 6-8 > 8 ans (\*)

Handicaps importants : moteur sensoriel intellectuel affectif (\*)

(\*) = entourez ce qui convient

Date de l'exercice : ..... / ..... / .....

Durée de l'exercice (approximativement) : ..... H ..... Min

#### Menu (Recettes, Dépenses, Etat du Porte-Monnaie)

Les termes utilisés sont-ils compris par la personne..... OUI NON

Les symboles utilisés sont-ils immédiatement compris par

la personne ? ..... OUI NON

La personne se base-t-elle sur les couleurs pour s'orienter ?..... OUI NON

La personne clique-t-elle sur FIN par inadvertance ?..... OUI NON



### Saisie d'une somme par souris

La personne reconnaît-elle facilement les billets et pièces ?.....	OUI NON
Y a-t-il confusion entre 10 fr et 10 c ou 20 fr et 20c (Suisse) ?....	OUI NON
Y a-t-il confusion entre 5 fr et 5 c (Suisse) ?.....	OUI NON
La personne remarque-t-elle qu'une trace apparaît lorsqu'elle a cliqué sur un billet ou une pièce ?.....	OUI NON
Arrive-t-il que la personne dépasse le nombre de billets ou pièces représentables sur l'écran ?.....	OUI NON
La personne n'est-elle pas perturbée que la trace n'apparaît plus lorsqu'elle a entré plus de billets ou pièces que ne le permet l'écran (si cela arrive) ?.....	OUI NON
Arrive-t-il que la personne clique sur un billet ou une pièce se trouvant sur la partie droite de l'écran (trace) ?.....	OUI NON
Arrive-t-il que la personne clique sur un autre billet (pièce) que celui (celle) désiré(e) par imprécision ?.....	OUI NON
La personne comprend-elle aisément la différence entre "quitter" et "fini" ?.....	OUI NON
La personne a-t-elle le temps de voir le thermomètre fluctuer ?...	OUI NON
La personne comprend-elle que cet écran sert à la fois à saisir une recette, une dépense ou la somme qu'elle détient actuellement ?.....	OUI NON
Arrive-t-il que la personne place mal le suiveur de la souris (flèche rouge) à cause de la forme du suiveur ?.....	OUI NON

### Saisie du moment d'une dépense

La personne comprend-elle les trois dessins affichés ?.....	OUI NON
La personne se base-t-elle sur les couleurs des dessins pour s'y retrouver ?.....	OUI NON
La personne se base-t-elle sur le texte pour s'y retrouver ?.....	OUI NON

### Saisie du jour d'une dépense

La personne se base-t-elle sur les couleurs des dessins pour s'y retrouver ?.....	OUI NON
--	---------

La personne se base-t-elle sur le texte pour s'y retrouver ?.....	OUI NON
Les couleurs utilisées ne sont-elles pas confondues avec les couleurs des postes budgétaires ?.....	OUI NON

#### Saisie du poste budgétaire d'une dépense

La personne comprend-elle les dessins ?.....	OUI NON
La personne se trompe-t-elle de poste budgétaire ?.....	OUI NON
La personne met-elle presque toutes ses dépenses dans le poste "divers" ?.....	OUI NON
La personne se base-t-elle sur les couleurs des dessins pour s'y retrouver ?.....	OUI NON
La personne se base-t-elle sur le texte pour s'y retrouver ?.....	OUI NON

#### Saisie d'une somme par clavier

La personne éprouve-t-elle des difficultés à taper un chiffre au clavier ?.....	OUI NON
Arrive-t-il régulièrement que la personne entre des caractères qui ne sont pas des chiffres ?.....	OUI NON
La personne est-elle perturbée que tout ce qu'elle a tapé s'efface lorsqu'elle entre des caractères qui ne sont pas des chiffres (si cela arrive) ?.....	OUI NON

#### Etat du porte-monnaie (par graphique)

La personne comprend-elle que la somme qu'elle voit est celle qu'elle a communiquée au programme au début de l'exercice ?.....	OUI NON
Pour la réglette du temps, le jour actuel est-il clairement perçu ?	OUI NON
Pour la réglette du temps, la couleur aide-t-elle à percevoir les jours passés et les jours à venir ?.....	OUI NON
La personne clique-t-elle sur des objets non-interactifs ?.....	OUI NON

- La personne comprend-elle que, selon les cas, on lui indique qu'elle doit encore enregistrer des dépenses ou des recettes pour mettre ses comptes à jour ?..... OUI NON
- La personne comprend-elle, lorsqu'on lui montre la liste des recettes et des dépenses (si ce paramètre a été choisi) que ses comptes sont à jour ?..... OUI NON

#### Etat du porte-monnaie (par nombre)

- La personne comprend-elle que la somme qu'elle voit est celle qu'elle a communiquée au programme au début de l'exercice ?..... OUI NON
- La personne comprend-elle, lorsqu'on lui montre la liste des recettes et des dépenses (si ce paramètre a été choisi) que ses comptes sont à jour ?..... OUI NON
- La personne comprend-elle que, selon les cas, on lui indique qu'elle doit encore enregistrer des dépenses ou des recettes pour mettre ses comptes à jour ?..... OUI NON

#### Présentation des recettes et des dépenses

- La personne comprend-elle qu'il s'agit de la liste des recettes et des dépenses ?..... OUI NON
- La personne a-t-elle le temps de voir le thermomètre fluctuer ?... OUI NON

#### Généralités

- Les demandes de confirmations sont-elles nécessaires à cette personne ?..... OUI NON
- Le son aide-t-il la personne à se situer ?..... OUI NON
- L'accent anglais est-il très gênant pour la compréhension ?..... OUI NON

#### 3) Explication de la grille

Tout d'abord, on trouve le nom de la personne. Ceci était nécessaire à notre vision des choses. En effet, nous voulions que ce type de questionnaire soit rempli après chaque session de travail avec chaque

utilisateur. Le but de cette demande était de voir s'il y avait moyen de voir à quel moment il n'y avait plus d'évolution dans les réponses à ce questionnaire. On peut en effet accepter qu'un certain nombre de choses ne soient pas comprises et que des erreurs soient commises pendant un certain temps correspondant à la durée d'apprentissage du logiciel. Les erreurs commises pendant cette période ne doivent pas nécessairement donner lieu à des modifications du logiciel car elles peuvent disparaître une fois que le logiciel sera dominé. Le nom est donc nécessaire puisqu'il faut pouvoir retrouver tous les questionnaires relatifs à la même personne afin de pouvoir déterminer son "moment de stationnarité". Ceci nous permettrait alors d'arriver à la conclusion qu'au maximum après cinq sessions (par exemple), les réponses ne changent plus et donc que, lorsqu'on exploitera les résultats, il ne faudra pas tenir compte des cinq premiers questionnaires. Ceci nous permettrait également de pouvoir dire, pour une prochaine évaluation du logiciel, que ce questionnaire ne doit être rempli qu'une seule fois après minimum cinq séances.

Le sexe, l'âge réel, l'âge de réalisation et les handicaps importants devraient permettre d'opérer une certaine classification des utilisateurs. Ceci est nécessaire pour l'exploitation des résultats car comparer les résultats de personnes tout à fait différentes n'a aucun sens (bien que vouloir classer les handicapés mentaux reste à notre avis une tâche assez hasardeuse).

La date de l'exercice nous donne la possibilité de retrouver la trace correspondant à un questionnaire. En effet, chaque fichier contenant la trace d'une session porte un nom de la forme suivante : [nom de l'utilisateur, numéro de la session, ".TRACE"], le numéro de la session augmentant de un à chaque nouvel exercice. On peut donc retrouver la chronologie des traces. Et, étant donné que l'on dispose de la date de chaque questionnaire, il est facile de mettre en correspondance chaque questionnaire avec sa trace.

La durée totale de l'exercice nous permet, sachant grâce à la trace ce qui a été fait durant la session, et sachant quels paramètres ont été choisis, de voir si l'exercice a eu une durée acceptable ou si l'utilisateur a éprouvé beaucoup de difficultés. En effet, il est possible que l'utilisateur ne fasse aucune erreur, mais mette une heure pour entrer une recette et une dépense, ce qui ne serait pas une bonne chose pour sa motivation, et ce qui

nécessiterait un effort de concentration trop important et le laisserait incapable de réaliser d'autres tâches après sa session.

On peut voir que toutes les questions sont à répondre par OUI ou NON, ce qui permet de compléter la grille relativement rapidement. De plus, il ne faut pas répondre à toutes les questions puisque le questionnaire est organisé en paragraphes dont chacun correspond à un type de dialogue choisi lors de la paramétrisation. Ainsi, si l'éducateur a choisi que l'utilisateur X devra enregistrer une somme par la souris, il ne devra pas répondre aux questions constituant le paragraphe intitulé "saisie d'une somme par clavier".

Il faut encore noter que cette grille a trait à une ancienne version du programme qui ne correspond plus aux spécifications que nous avons présentées dans le chapitre 4. De plus, ce questionnaire est la version condensée ; la version complète contient en plus des espaces réservés aux observations et suggestions où l'éducateur note diverses choses dont des solutions éventuelles à des carences qu'il a pu remarquer. La version complète est remplie une seule fois par l'éducateur ( ou plus s'il le juge nécessaire). C'est ainsi par exemple que nous avons vécu la situation suivante. Le thermomètre de couleur fonctionnait dans l'ancienne version comme ceci : il se remplissait toujours de couleur rouge, ne comportait pas de repère jaune et représentait toujours la somme actuellement saisie, c'est-à-dire que le niveau rouge montait qu'il s'agisse d'une recette ou d'une dépense. Le laps de temps dont nous disposions pour la construction de la grille ne nous avait pas permis d'expliquer en profondeur aux handicapés le fonctionnement du thermomètre et la seule question à ce sujet apparaissant dans la grille était : "La personne a-t-elle le temps de voir le thermomètre fluctuer ?". La réponse était très souvent négative, ce qui nous faisait croire qu'il faut faire monter le niveau du thermomètre moins rapidement. Mais l'espace libre laissé pour les remarques a permis à un éducateur de nous préciser que l'utilité même de ce type de thermomètre était à remettre en cause parce que les handicapés ne pouvaient comprendre quoi que ce soit à ce type de feed-back. Avec un éducateur, nous avons alors cherché des solutions et nous sommes arrivés, par une nouvelle itération du processus de développement, au fonctionnement actuel que nous avons présenté dans le chapitre 4 (ce nouveau type de thermomètre n'a pu être évalué à ce jour).

#### 6.4. Les expérimentations

Nous allons voir ici les expérimentations qui ont été menées et nous donnerons à titre indicatif, les résultats que nous avons tirés des traces de manipulations et des questionnaires.

La première expérimentation s'est déroulée à La Castalie avec un groupe d'adolescents handicapés mentaux. Parmi ces jeunes se trouvaient ceux avec qui nous avons travaillé pendant la phase de spécification. Ils ont donc utilisé le programme lors de certaines de leurs séances de classe. L'utilisation fut périodique et tous les jeunes n'ont pas utilisé le programme un même nombre de fois.

Huit jeunes ont utilisé le programme, mais deux d'entre eux ne l'ont fait qu'une fois et un autre, deux fois. Nous avons décidé que ces traces n'étaient pas exploitables. Par conséquent, nous avons gardé celles de Michel (4 utilisations), Sylvie (4), Jean-Claude (5), Fabienne (9) et Ghislaine (10). En réalité, ces 5 personnes avaient déjà utilisé le programme au cours de notre second stage, et c'est au cours de ces séances que nous avons mis en application la méthode pour construire les questionnaires, que nous avons exposée à la section 6.3.2.

##### 6.4.1. L'exploitation des traces

Nous avons procédé en deux étapes :

- la première consistait, à partir des traces (que nous avons récupérées, grâce aux copies de disquettes que nous ont envoyées les éducateurs de La Castalie) à reconstituer la chaîne des états par lesquels le programme était passé lors de chaque session de travail.

- ensuite à relever les fréquences, sur toutes les utilisations cumulées pour un utilisateur, des transitions critiques, c'est-à-dire, par exemple, si un utilisateur a enregistré 10 recettes, combien de fois il a quitté cette fonction avant d'avoir fini.

Remarques : - bien entendu, on peut relever les fréquences au sein d'une seule session, mais nous avons fait l'étude sur les utilisations cumulées, pour ne pas avoir trop de fréquences nulles.

- de plus, il faut ajouter que ces calculs ne devraient porter que sur les utilisations où la personne handicapée domine le programme, c'est-à-dire quand elle a atteint son niveau de stationnarité.

Les traces, les chaînes des états, les calculs des fréquences et une première interprétation de ces résultats sont dans les annexes.

Voyons quelques points qui peuvent attirer l'attention quand on regarde les résultats de ces traces (tout en sachant bien qu'aucune conclusion ne peut être tirée d'un si petit nombre de tests).

\* Pour tous les utilisateurs réunis :

il est assez frappant de constater que tous les utilisateurs commencent majoritairement par enregistrer une dépense. Toutefois, n'oublions pas que souvent il s'agissait d'exercices plus ou moins dirigés par l'éducateur, donc il n'est pas du tout certain que cela relève de choix posés par la personne handicapée elle-même. Ensuite, il est arrivé à 3 des 5 utilisateurs de cliquer sur FIN dans le menu (pour clôturer la session) puis d'infirmier cette demande. Cela nous a amené à modifier la présentation graphique du menu ; nous avons réduit les dimensions de la case FIN et surtout, nous l'avons éloignée des trois autres (car elle est sémantiquement différente). En effet dans la version initiale, elle était juste sous les trois autres icônes.

\* Pour Ghislaine :

- au niveau de l'enregistrement de la somme initiale, sur 10 séances, elle a demandé une seule fois d'effacer. Cette fois là, elle utilisait le clavier, mais elle l'a utilisé à 8 autres reprises sans erreurs. Donc ce n'est pas le type de dialogue qui est à mettre en cause, il s'agit sans doute d'une erreur de frappe comme tout le monde en fait.

- au niveau de l'enregistrement des recettes, sur 14 fois, elle a quitté une fois. Cela s'est passé dans la toute première session de travail. Ensuite elle est allée regarder l'état du porte-monnaie après quoi elle a réellement enregistré une recette. Il ne semble pas que ce soit une erreur qui démontre une incompréhension due à l'interface.

- au niveau des dépenses, tout est allé très bien pour enregistrer la somme, puis pour enregistrer le moment de la dépense, elle s'est trompée une fois (elle avait donné le "mardi" et a corrigé par le "lundi"). Un seul cas

s'étant produit, il semble de nouveau que ce ne soit pas grave. Bizarrement, 8 fois sur 26 le poste budgétaire a été infirmé. En fait cela s'est passé dans 2 sessions ; à chaque fois Ghislaine a donné un poste, l'a infirmé, en a donné un autre ..., 4 fois de suite. Donc il ne s'agit pas d'erreurs se produisant à chaque session de travail, mais de 2 moments dans 2 sessions seulement.

Nous ne savons pas très bien qu'en dire (dans un tel cas l'apport d'une discussion avec l'éducateur de Ghislaine est évidemment très grand). En tout cas, pour nous cela ne signifie pas qu'elle a des problèmes à reconnaître les icônes des postes budgétaires, sinon les erreurs seraient diffusées dans toutes les sessions. Il s'agit peut-être d'un amusement, ou d'une journée où elle avait des difficultés à poser un choix ou à se concentrer.

Lorsque nous disons qu'elle n'a pas de problèmes à reconnaître les icônes, notre subjectivité joue, parce que nous connaissons Ghislaine, pour avoir travaillé avec elle à toutes les phases du projet. Mais le côté humain est tellement important, qu'il n'est absolument pas opportun de vouloir faire abstraction des expériences, des relations que nous avons eues avec les personnes handicapées. Donc si cela peut nous aider à interpréter les résultats, c'est tant mieux.

Il reste à dire que pour les trois types d'enregistrements (somme initiale, recettes et dépenses), quand elle appuie sur FINI, elle confirme toujours, donc elle semble bien maîtriser sa tâche et bien réaliser quand elle a fini

- en fin de session, une fois sur 11 elle a infirmé sa demande de FIN, elle fait donc partie des trois utilisateurs dans le cas, dont nous avons parlé avant.

Donc ceci termine les observations des traces de manipulations de Ghislaine, et on voit que cela peut déclencher des idées et nécessiter des rencontres avec les éducateurs pour approfondir certains points.

\* Pour Fabienne :

- au niveau de l'enregistrement de la somme initiale, elle a demandé 3 fois d'effacer (ce qui reste peu sur un total de 9 séances), mais dans un cas particulier, cela a attiré notre attention. En effet, elle a demandé d'effacer, a confirmé, mais juste après, elle a demandé de recommencer tout (c'est-à-dire d'effacer tout ce qu'elle avait enregistré jusqu'alors). Cela peut



signifier qu'après avoir effacé la dernière pièce de monnaie entrée, elle a remarqué (ou l'éducateur lui a fait remarquer) qu'il y avait d'autres erreurs, et donc a demandé de tout recommencer, ou bien cela peut signifier qu'elle s'est trompée en demandant d'effacer et qu'en fait elle voulait directement tout recommencer (ce qui voudrait dire qu'il y a confusion entre l'icône "EFFACER" et l'icône "RECOMMENCER"). Bien sûr une seule occurrence de cette séquence ne suffit pas, mais si un grand nombre d'utilisateurs faisaient la même chose, cela signifierait sans doute qu'il faut changer les icônes.

En fait lorsque l'expérimentation a été menée, les icônes "EFFACER" et "RECOMMENCER" étaient 2 rectangles, par la suite pour mieux les différencier, nous avons donné une forme ovale à l'un d'eux. De même dans les fonctions d'enregistrement de recettes et de dépenses (où il n'y a pas d'icône "RECOMMENCER"), nous avons donné la forme ovale à l'icône "QUITTER".

- au niveau de l'enregistrement des recettes, elle a quitté une fois sur 14. Donc rien de particulier là dedans.

- pour les dépenses, elle a quitté 3 fois, mais 3 fois elle a repris la même fonction, ce qui signifie qu'elle s'était trompée dans ses manipulations mais pas dans le choix de la fonction dans le menu. Nous avons aussi remarqué que la seule fois où elle a demandé d'effacer sa dernière pièce enregistrée, c'était pour remplacer une pièce de 5 francs par une de 5 centimes. Nous avons parlé de cette éventualité, mais comme c'est le seul cas que nous ayons rencontré, nous n'estimons pas qu'il y ait besoin (pour l'instant) de changer la présentation des 2 icônes.

Remarquons ensuite que Fabienne confirme toujours ses demandes de fin de fonction, ce qui pourrait signifier qu'elle aussi, a une bonne idée de la fin d'une tâche.

Elle confirme aussi toujours postes budgétaires et moments des dépenses.

- en fin de session, elle fait aussi partie des 3 utilisateurs qui ont infirmé une demande de FIN (elle l'a fait 3 fois sur 12). Rappelons que la forme du menu a été modifiée depuis pour bien différencier cette icône des trois autres fonctions.

\* Pour Sylvie :

- Les traces de Sylvie sont remarquables, pour les fonctions

d'enregistrement des recettes et des dépenses, elle ne demande jamais d'effacer ou de quitter, elle confirme toujours ses postes budgétaires et les moments des dépenses. Par contre au niveau de l'enregistrement de la somme initiale, il y a de gros problèmes. En effet, sur 4 séances, elle a demandé 12 fois d'effacer et 11 fois de recommencer tout. Mais en réalité la grande majorité de ces "erreurs" proviennent de la dernière séance : au cours de l'enregistrement de la somme initiale, elle a demandé 7 fois d'effacer et 8 fois de recommencer. C'est énorme ! Cela méritait de plus amples explications, nous les avons eues auprès des personnes qui entouraient Sylvie à l'époque. En fait, elle était très perturbée par son départ imminent de La Castalie (elle allait atteindre la limite d'âge).

Ceci est un exemple de l'importance des aspects affectifs dans ce type de projet.

Pour les autres séances, Sylvie s'est très bien comportée, tout en commettant plus d'erreurs lors de la saisie de la somme initiale que lors des enregistrements des recettes et des dépenses. De nouveau, ceci peut être imputé à des aspects affectifs, qui font qu'elle a quelques difficultés au début d'une session à "entrer" dans le travail, puis s'y intègre très bien.

\* Pour Michel :

Ses traces de manipulations, montrent un minimum d'erreurs, elles se situent uniquement dans l'enregistrement de la somme initiale (et jamais plus d'une fois par session). Ceci indiquerait peut-être aussi une petite difficulté à entrer dans son nouveau travail. En tous cas, les résultats semblent excellents en ce qui concerne Michel, qui est très timide et semble apprécier le "contact" avec l'ordinateur. Bien sûr nous ne pouvons aller plus loin dans l'évaluation des apports pédagogiques du programme, c'est une tâche de la compétence des psychologues.

\* Pour Jean-Claude :

Ici aussi très peu d'erreurs. Il a demandé d'effacer une seule fois lors de la saisie de la somme initiale, on ne peut donc rien en conclure. Lors d'enregistrements de dépenses, il a effacé à 2 reprises et quitté la fonction 2 fois (sur 13). Il n'y a là rien de particulier, et le taux d'erreur n'est pas alarmant. Donc, il semble que pour Jean-Claude aussi l'utilisation du programme ne démontre pas d'incohérences graves, ou d'incompatibilité avec ses capacités.

## 6.4.2. Exploitation des questionnaires

### 6.4.2.1. L'analyse statistique

Il faut avant tout noter que la phase d'expérimentation n'est pas terminée à ce jour. Nous avons récolté et exploité jusqu'à maintenant 19 questionnaires (ayant trait à l'interface uniquement) pour 6 personnes et couvrant une période de un mois et demi. Nous présentons, dans les annexes, 4 questionnaires complets, relatifs à l'expérimentation menée auprès de quatre personnes adultes, à l'Institution Villa Allegro de Spa.

Il est clair que l'exploitation nécessite de bien plus grandes quantités de résultats si on veut pouvoir tirer des conclusions, et surtout des questionnaires relatifs à un plus grand nombre d'individus. Cette nécessité est encore accentuée par deux facteurs : un premier lié au fait que les utilisateurs sont handicapés, donc la population étudiée est très hétérogène, l'autre ayant rapport à la haute paramétrisation du programme.

\* Il faut pouvoir effectuer une certaine classification des utilisateurs si l'on veut pouvoir comparer les résultats obtenus. En évaluant le programme par classe d'utilisateurs, on devrait pouvoir examiner quels sont les problèmes liés à chaque classe et, éventuellement, rejeter certaines classes qui éprouvent trop de difficulté de compréhension. Cela nous permettrait de répondre plus précisément à la question : "A quelle population le projet est-il destiné ?" (voir section 4.1.1.3.)

\* Deuxièmement, il faut bien se rendre compte que ce programme est très paramétré, ce qui signifie qu'au sein de chaque classe d'utilisateurs, il serait nécessaire de définir des sous-classes selon les paramètres de dialogue qui ont été choisis. C'est au sein de ces sous-classes que l'on va pouvoir comparer les résultats et en tirer des conclusions intéressantes.

Si l'on forme ainsi des groupes dans lesquels les individus sont plus ou moins similaires (échantillons homogènes), on peut, au sein de chacun d'entre-eux, expérimenter le programme et exploiter les réponses aux questionnaires par la statistique inférentielle. Si les questionnaires sont complétés après chaque session, il ne faudra garder que le dernier, c'est-à-dire celui qui correspond à une maîtrise du logiciel par l'utilisateur

(phase d'apprentissage terminée). La première chose à faire est de calculer, pour chaque groupe, la fréquence des OUI (et des NON) pour chaque question. Si l'on fait l'hypothèse que les 18 questionnaires que l'on a reçus proviennent de 18 utilisateurs différents, on obtient des fréquences dont quelques-unes sont présentées ci-dessous (le tableau complet des fréquences étant présenté en annexe). Les questions sont numérotées dans l'ordre d'apparition dans la grille.

<u>Question</u>	<u>total réponses</u>	<u>total OUI</u>	<u>total NON</u>	<u>% OUI</u>	<u>% NON</u>
3	15	3	12	20	80
13	12	10	2	83,3	16,4
18	12	4	8	33,3	66,7
20	15	4	11	26,7	73,3
23	18	18	0	100	0
26	18	4	14	22,3	77,7
36	3	0	3	0	100
39	5	0	5	0	100

Il s'agit de calculer à partir de quand une fréquence observée est significative ( ex : région critique  $R_c = [ P \geq 0.8 ]$  ) et déterminer la fréquence à partir de laquelle on rejette. Si l'on avait une distribution normale (c'est-à-dire qu'il nous faudrait alors 100 ou 200 observations par groupe), on pourrait calculer un intervalle de confiance. Mais il est clair que l'on ne récoltera toujours qu'un nombre relativement petit de questionnaires par groupe (de l'ordre d'une vingtaine), ce qui nous donne donc une distribution binomiale, chaque question ne pouvant recevoir que deux réponses OUI ou NON (succès et échec ou l'inverse, selon le type de question). Or, on sait qu'on trouvera un intervalle de confiance pour une binomiale en consultant les tables (cette table est présentée en annexe).

Ainsi, appelons :

w : l'échantillon

n : l'effectif de l'échantillon (nombre de réponses à chaque question)

S : le nombre de succès (nombre de OUI ou de NON à chaque question, selon le type de question)

p : la probabilité d'un succès

Alors, le nombre de succès S, sur n observations, a une distribution binomiale de signature (n,p) :  $\Pr [ S (w) = k ] = C_n^k p^k (1-p)^{n-k}$ .

Si la valeur observée de  $S = i$ , il suffit de poser (pour un niveau d'incertitude  $\alpha = 0.05$ ) :

$$F_{B(n, p2)}(i) = 0.025$$

$$F_{B(n, p1)}(i) = 0.975$$

et la consultation des tables nous fournit les valeurs  $p1$  et  $p2$ , ce qui détermine un intervalle de confiance pour  $p$  :  $p1 \leq p \leq p2$  ( $\Pr [ p1 \leq p \leq p2 ] = 1-0.05$ ). L'utilisation de la table se fait de la façon suivante : en abscisse, on trouve le nombre de succès observés divisé par l'effectif de l'échantillon (c'est-à-dire, pour nous, la fréquence de OUI ou de NON à chaque question). Etant donné l'effectif de l'échantillon, on rencontrera alors deux courbes portant ce nombre. Pour autant que la fréquence observée ne dépasse pas 50 %, la courbe la plus basse donne en ordonnée la limite inférieure de l'intervalle de confiance et la plus haute en donne la limite supérieure (si la fréquence est de plus de 50 %, on cherchera les limites pour  $f' = 1 - f$ , et on calculera  $p1 = 1 - p'1$  et  $p2 = 1 - p'2$  ( $p'1$  et  $p'2$  trouvées dans la table), ce qui fait que c'est cette fois la courbe la plus haute qui détermine la limite inférieure de l'intervalle de confiance).

Les exemples suivants nous permettront de mieux comprendre en quoi cette méthode peut nous permettre de relativiser les réponses obtenues aux questionnaires.

Ainsi, la question 13 ("La personne comprend-elle aisément la différence entre "quitter" et "fini" ?") nous donne une fréquence de 83,3 % de OUI, ce qui constitue en lui-même un score acceptable. Cependant, si l'on consulte les tables, on trouve l'intervalle de confiance  $0.6 \leq p \leq 0.97$ , ce qui n'est pas suffisant pour affirmer que la distinction entre ces deux

notions est clairement perçue ; il peut donc être nécessaire de chercher une autre solution.

Il faut cependant dire que l'intervalle de confiance donnera lieu à des conclusions différentes selon le type de question. En effet, il est possible que la seule chose que l'on désire est de vérifier que l'on n'ait pas une proportion nulle. Dès lors, si l'intervalle de confiance ne recouvre pas la valeur 0 % et même s'il est très grand, il permettra de considérer que la fréquence observée est favorable. Ainsi, les questions 3, 18, 20 et 26, qui prennent en compte la couleur des icônes, donnent un score moyen de 25 % de OUI, ce qui nous indique que 25 % des utilisateurs se basent sur les couleurs pour s'orienter. L'intervalle de confiance correspondant à la fréquence observée est  $0.07 \leq p \leq 0.52$  et ne contient donc pas la fréquence 0 %, ce qui nous montre que les différentes couleurs des icônes ne sont de toute façon pas inutiles et qu'elles peuvent être gardées. On pourra arriver à des conclusions similaires lorsque le but est de tester que la fréquence est toujours supérieure ou inférieure à 50 %.

Il est bien entendu que les scores extrêmes 0 % -100 % ne nécessitent normalement pas une telle analyse et que les conclusions sont immédiates (pour un score de 0 % et un effectif de 20, on trouve l'intervalle  $0 \leq p \leq 0.17$ ). Par exemple, la question 23 ("La personne comprend-elle les dessins représentant les postes budgétaires ?") donne 100 % de OUI, preuve qu'il n'est pas nécessaire d'aller plus loin dans la recherche de pictogrammes représentant ces différents postes. Inversément, les questions 36 et 39, qui sont en fait la même question mais pour des dialogues différents ("La personne comprend-elle, lorsqu'on lui montre la liste des recettes et des dépenses, que ses comptes sont à jour ?") donne une fréquence de 100 % de NON, ce qui indique qu'il existe un réel problème. Néanmoins, il faut faire attention que l'effectif pour ces questions était de 8 et que dans ce cas, l'intervalle de confiance pour 0 % de OUI est  $0 \leq p \leq 0.37$  (bien entendu, dans ce cas, un score de 37 % ne change rien à la conclusion, mais ce ne sera pas toujours le cas).

#### **6.4.2.2. L'analyse en composantes principales**

Les utilisateurs qui ont expérimenté le programme peuvent être

représentés par un certain nombre de variables, dont chacune correspond à une question de la grille d'évaluation (variables à deux valeurs : 0 ou 1). Ainsi, si la grille contient 50 questions, chaque individu peut être représenté par un point dans un espace à 50 dimensions. Il est très difficile alors, de comparer les individus représentés par un nombre aussi important de variables.

Le but de l'analyse en composantes principales est double :

- d'une part, elle étudie les corrélations entre les différentes questions (les axes initiaux de l'espace), afin de déterminer des combinaisons linéaires de questions présentant une corrélation assez importante et ainsi construire de nouveaux axes (appelés axes factoriels) dont le nombre est inférieur à celui des axes initiaux

- d'autre part, la méthode situe les individus par rapport à ces axes factoriels, ce qui permet de comparer les utilisateurs.

Dans le rapport RP10/89 [M. Noirhomme-Fraiture, 1989], un exemple concernant nos questionnaires, a été traité par cette méthode. Il reprend le dernier questionnaire de 5 utilisateurs. Il a fallu garder uniquement les questions communes à tous (c'est-à-dire relatives à des éléments de dialogue qu'ils ont tous expérimenté), puisqu'il faut que les utilisateurs soient caractérisés par un même ensemble de variables. De plus, les questions sans réponse n'ont évidemment pas été prises en compte, ainsi que celles qui recevaient les mêmes réponses pour tous les utilisateurs (car dans ce cas la variance est nulle et la méthode ne traite pas de variance nulle). Les questions restantes sont les suivantes :

Q1 : Les termes utilisés sont-ils compris par la personne ? (Menu)

Q14 : La personne a-t-elle le temps de voir le thermomètre fluctuer ?  
(Saisie d'une somme)

Q19 : La personne se base-t-elle sur le texte pour s'y retrouver ?  
(Dépense)

Q21 : La personne se base-t-elle sur le texte pour s'y retrouver ?  
(Moment de la dépense)

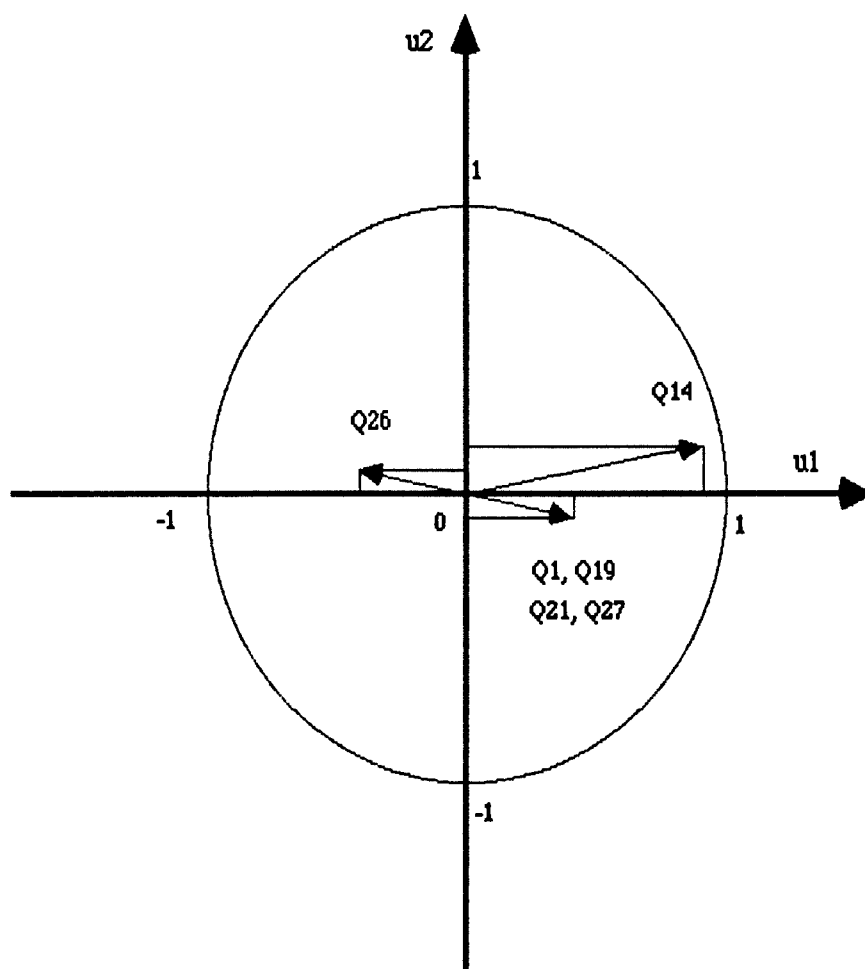
Q26 : La personne se base-t-elle sur la couleur des dessins pour s'y retrouver ?

Q27 : La personne se base-t-elle sur le texte pour s'y retrouver ?  
(Postes budgétaires)

L'application de la méthode a déterminé 2 axes factoriels, donc toute l'information contenue dans les 6 axes initiaux (correspondant aux 6 questions retenues) se retrouve dans ces 2 axes.

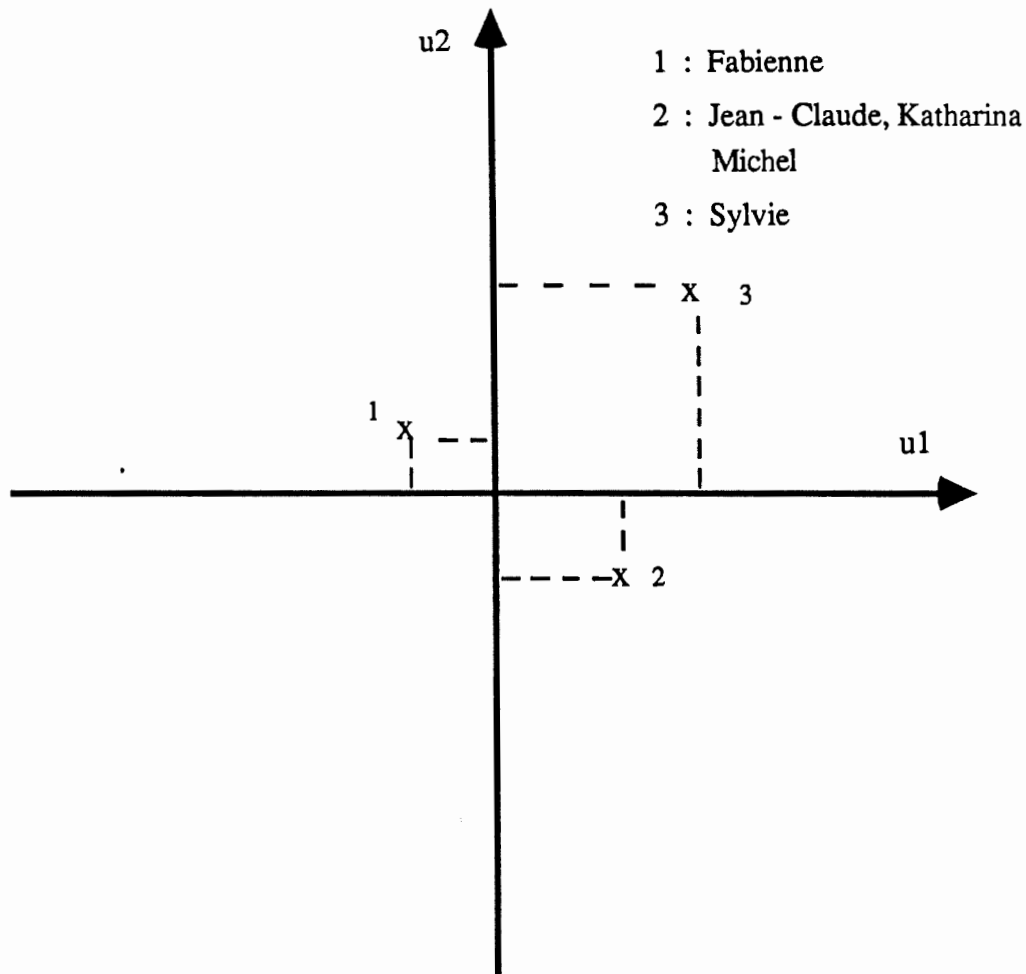
Le premier axe oppose la compréhension par texte à la compréhension par couleur et le deuxième est très lié à l'appréciation de la fluctuation du thermomètre (corrélation = 0.99).

On peut dès lors dessiner un graphe représentant les 6 axes initiaux (dont 4 sont confondus) et les 2 axes factoriels  $u_1$ ,  $u_2$ .





On peut également replacer les utilisateurs dans cet espace à deux dimensions ( $u_1, u_2$ ).



Ainsi, on voit que le premier axe oppose Fabienne aux autres utilisateurs ( $u_1 < 0$  pour Fabienne, et  $u_1 > 0$  pour les autres), et que le deuxième oppose Sylvie à Jean-Claude, Katharina et Michel.

L'expérimentation que nous avons menée avec ces 5 personnes nous a en effet montré que Fabienne n'a presque aucun repère en lecture et il est donc normal qu'elle se repère à l'aide des couleurs, contrairement aux autres. De même, Sylvie, bien qu'ayant des problèmes comportementaux très prononcés, montre de très bonnes qualités intellectuelles. Il n'est pas étonnant qu'elle ait cherché à comprendre ce que signifiait ce thermomètre, alors que les autres n'y faisaient pas attention.

On voit donc que la méthode arrive à des conclusions cohérentes par

rapport à la réalité observée. Il faut ajouter que dans l'exemple traité dans ce rapport, il aurait été facile d'arriver aux mêmes conclusions en observant simplement les réponses aux questions. Cet exemple n'est donc proposé qu'à titre d'illustration de la méthode et il est clair que celle-ci prend tout son intérêt lorsque le nombre d'individus et de variables est plus grand, ce qui rend l'analyse intuitive impossible.

## Chapitre 7. Conclusions et perspectives

Pour conclure, nous allons tout d'abord synthétiser les différentes phases de notre démarche tout au long de ce projet. En particulier la phase de spécification et celle de l'évaluation du programme. Enfin, nous verrons quelques perspectives pour des travaux ultérieurs, tout d'abord en ce qui concerne le programme lui-même et ensuite de façon plus générale, en ce qui concerne le domaine de l'informatique et du handicap mental et la façon de mener des projets de ce genre.

### 7.1. Phase de spécification du projet

Nous avons vu que la démarche suivie était essentiellement itérative et que nous avons décidé de spécifier en parallèle les fonctionnalités et l'interface. Nous restons convaincu qu'il n'est pas souhaitable de dissocier ces deux spécifications. De plus, la démarche itérative est indispensable, car ce qui paraissait fondé lors de discussions avec les éducateurs pouvait parfois être tout à fait rejeté par les futurs utilisateurs.

Ensuite, nous avons vu que la spécification de l'interface pouvait être divisée en deux parties; la spécification de l'interface abstraite (la tâche) et la présentation concrète de celle-ci. S'il est vrai que nous n'avons pas complètement spécifié la tâche avant de nous préoccuper de la présentation, tout un travail, à ce niveau, avait déjà été effectué lors des groupes de travail organisés par le centre PSINHA. Ces groupes avaient déjà analysé et étudié les processus habituellement utilisés par les personnes handicapées mentales pour la gestion de l'argent.

C'est sur base de ce travail déjà accompli en Belgique que nous avons mené différents tests avec les personnes handicapées.

Il est certain qu'une spécification complète et rigoureuse de la tâche avant tout travail sur la présentation, ne peut être que bénéfique. Toutefois la recherche se situait prioritairement au niveau de la présentation.

En conclusion, si nous n'avons pas fait preuve d'une rigueur absolue lors de la spécification de l'interface, cela n'a eu aucun effet préjudiciable .

Quant au temps nécessaire pour assurer cette collaboration, nous

pensons que le mois que nous avons eu pour faire les spécifications en collaboration avec les psychologues et les jeunes était nécessaire (et même un peu plus de temps n'aurait pas nuit).

C'est réellement une phase trop importante pour ne pas y consacrer le temps nécessaire. Nous savons qu'en général, pour le développement de logiciels, trop peu de temps est accordé à la phase de spécification, ce qui reporte beaucoup de travail à la maintenance. Ce qui est vrai pour les logiciels standard l'est tout autant (si pas plus) quand le programme est destiné à des personnes ayant un handicap mental.

En conclusion, cette phase de spécification doit se faire sans aucun doute en collaboration avec les psychologues et les personnes handicapées. Elle doit être itérative et ceci implique donc qu'il faut y consacrer un certain temps. Il sera profitable pour la partie d'écriture du programme, qui est, dans le fond, secondaire et ne devrait pas être la préoccupation première, comme c'est hélas trop souvent le cas.

## 7.2. Phase d'évaluation

Nous avons vu que les deux moyens qui ont été mis en oeuvre pour évaluer le programme étaient d'une part les traces gardées après chaque session de travail, et d'autre part les questionnaires remplis par les éducateurs qui encadrent les personnes handicapées. Voyons ce qu'il y a à dire sur ces deux techniques complémentaires.

### 7.2.1. Les traces de manipulations

L'idée des traces est intéressante, toutefois leur exploitation est loin d'être aisée et de révéler rapidement des anomalies ou des inadéquations entre le programme et les utilisateurs. Relever les chaînes des états par lesquels l'utilisateur est passés lors d'une session de travail, calculer les fréquences de passage aux transitions intéressantes prend du temps et parfois cela n'apporte pas grand chose de nouveau. Nous verrons à la section 7.3. comment simplifier ces tâches, en les faisant faire par le programme. En tout cas, le contenu des traces, devrait sans doute être enrichi. Par exemple il n'y a aucune notion de temps, c'est-à-dire combien

de temps un utilisateur met-il pour réaliser une étape, pour poser un choix...etc. De même quand il tape quelque chose au clavier et qu'il efface puis corrige, on ne sait pas ce qu'il a effacé, or c'est sans doute intéressant de le savoir, pour voir quel type d'erreur il commet (s'il confond telle ou telle touche, s'il place la virgule au mauvais endroit dans un nombre...etc) et quel apprentissage pourrait permettre d'éviter ces fautes.

Donc, le contenu devrait sans doute être enrichi, et nous avons vu que le développement des traces entre dans le même processus itératif que celui qui encadre toutes les phases du projet.

Nous avons dit également que les traces pourraient servir de "déclencheur" en attirant l'attention sur des fautes commises. En réalité, la plupart des modifications qui ont été apportées au programme, au cours des "itérations", émanaient directement de discussions avec les éducateurs ayant employé le programme dans les institutions. Donc ceci tend à montrer l'importance des communications informelles dans ce type de projet.

Toutefois, il n'en reste pas moins vrai qu'il faut garder des traces des manipulations effectuées par les utilisateurs lors des sessions de travail. Car comme nous l'avons vu, elles peuvent mettre en lumière des faiblesses du programme et aussi elles permettent aux éducateurs de laisser les personnes handicapées seules face à l'ordinateur et de voir après coup, ce qu'elles ont fait (pas à pas), dans le but d'évaluer les résultats au niveau pédagogique de l'utilisation du programme, c'est-à-dire, par exemple, voir si cela permet d'acquérir certaines notions qui n'avaient pas été comprises par d'autres moyens pédagogiques. Ensuite, les traces, contrairement aux questionnaires, ne demandent aucun effort de la part des accompagnants, c'est une raison de plus pour les développer. Enfin, c'est le seul moyen qui permet de relever les critiques "implicites" des utilisateurs, puisque, contrairement aux applications classiques, ceux-ci ne font pas de critiques explicites.

### 7.2.2. Les questionnaires

Cette méthode est évidemment plus classique et nous avons vu qu'elle était employée par exemple pour évaluer l'emploi de techniques nouvelles pour aider les personnes I.M.C.. Mais il faut voir les deux tendances

extrêmes dans ce domaine. Soit le questionnaire est très court et très simple à remplir, mais contiendra sans doute peu d'informations intéressantes, soit il est très complet, mais demande beaucoup de temps pour être complété.

Quand nous avons élaboré le premier questionnaire sur l'interface homme-machine, qui rappelons-le devait être complété par l'éducateur après chaque session de travail, nous avons cherché à simplifier la tâche de la personne qui répond, pour cela les questions proposaient toutes le choix "OUI", "NON" pour répondre, et de plus il suffisait d'entourer la bonne réponse. Toutefois à cause de cette simplification, les informations récoltées perdaient un peu de richesse. En effet, certaines questions auraient été plus pertinentes si elles avaient permis une graduation plutôt que les choix extrêmes "OUI", "NON" : par exemple, pour la question concernant la synthèse vocale "L'accent anglais est-il très gênant pour la compréhension ?", il est possible que la personne comprenne sans problème un certain nombre de messages et peu ou pas du tout les autres ; dès lors, ni la réponse "OUI", ni la réponse "NON" ne donnent une bonne idée de la réalité. Or il est très important de savoir quels messages sont difficiles à comprendre et doivent être modifiés. Donc, on voit que la recherche de simplicité a tendance à appauvrir les informations que le questionnaire permet de recueillir. Donc, une échelle de 0 à 10 ou de 1 à 5 pourrait être proposée (il faut veiller à avoir une valeur médiane).

D'autre part, le deuxième questionnaire complet permet de recueillir beaucoup d'informations, mais il faut beaucoup de temps pour le compléter. Notons que ceci ne posera pas toujours des problèmes, si les éducateurs participant à l'expérimentation sont motivés par le projet (ce qui fut le cas dans la plupart des institutions avec qui nous avons collaboré).

### 7.3. Perspectives au niveau du programme

Chaque fois que nous avons rencontré des éducateurs qui avaient utilisé le programme, ils nous ont proposé, outre des critiques visant à améliorer ce qui existait, des idées nouvelles tant au niveau des fonctionnalités que des dialogues.

Le développement du programme dépend beaucoup de l'évaluation

qui en est faite. Si les dialogues s'avèrent insatisfaisants pour un grand nombre de personnes handicapées, ils devront avoir la priorité dans des travaux futurs, s'ils semblent corrects, on peut alors songer à élargir le champ des fonctionnalités.

Ceci étant dit, nous pouvons donner les propositions que nous avons relevées au cours des mois de travail mais que nous n'avons pas eu le temps de réaliser nous-mêmes. Ces propositions, du point de vue informatique, ne devraient pas poser de problèmes majeurs, ni demander trop de temps.

- Nous avons vu que lorsqu'on enregistre une somme en cliquant sur les billets et pièces, on voit apparaître un feed-back graphique de tout ce qui a déjà été enregistré (ce sont donc les petits dessins des pièces et des billets qui apparaissent sur le côté droit de l'écran). Au-delà de 5 billets et 7 pièces, plus rien n'apparaît car on a atteint les limites de l'écran. Une proposition serait d'ajouter en bout de ligne un nombre correspondant au nombre de fois qu'un élément a été sélectionné. Même si les nouveaux dessins n'apparaissent plus, ce nombre augmenterait.
- Dans le même ordre d'idée, il se peut que la somme maximale que représente le thermomètre rempli, soit dépassée. Actuellement, quand cela arrive, le thermomètre se remplit, mais rien n'indique le dépassement. On peut imaginer un système où dans ce cas, le thermomètre se remplirait dans une autre couleur, de manière à faire comprendre que quelque chose d'exceptionnel se passe.
- Il a été proposé de pouvoir effacer plus que le dernier billet ou la dernière pièce sélectionnée, c'est-à-dire de pouvoir "remonter" dans l'ordre inverse tous les billets et pièces sélectionnés.
- Concernant l'intérêt pédagogique du programme, il serait peut-être intéressant de permettre à l'utilisateur une approche de l'addition, de la multiplication et de l'écriture sous forme de compte classique. Par exemple, en ayant dans la partie droite de l'écran, non plus les traces graphiques, mais la retranscription sous forme de multiplication et d'addition des informations entrées. Ce qui pourrait donner la présentation suivante :

4	X	100	=	400
2	X	50	=	100
0	X	20	=	0
*	*	*	*	*
3	X	1	=	1

---

TOTAL	501	FRS
-------	-----	-----

Bien entendu, ce type de dialogue serait à choisir grâce à un paramètre.

- Le scénario d'utilisation actuel prévoit que l'utilisateur donne son nom, puis communique la somme qu'il a dans son porte-monnaie. Il peut ensuite choisir les différentes fonctions dans le menu. Une autre approche consisterait à commencer par introduire les dépenses et les recettes, le programme calculant, en fonction du solde de la session précédente, la somme que l'utilisateur est sensé avoir dans son porte-monnaie.
- Au niveau des traces, nous avons vu qu'elles peuvent être améliorées ; par exemple, le travail de reconstitution de la chaîne des états par lesquels le programme est passé, est assez fastidieux à faire. Dès lors, la trace devrait, outre sa forme textuelle, donner cette chaîne finie d'états. Il devrait ensuite calculer automatiquement les fréquences cumulées (sur toutes les sessions d'un même utilisateur) de passage aux transitions intéressantes. Ceci est tout à fait réalisable et simplifierait considérablement le travail d'exploitation des traces.
- Une autre proposition intéressante concernant les traces serait d'offrir un outil à l'éducateur pour analyser ces traces et pouvoir faire des sélections : par exemple, si l'éducateur veut contrôler ce que l'utilisateur dépense pour la nourriture, un programme analyserait les traces et en extraierait uniquement l'information relative à ce poste budgétaire.
- Nous avons déjà dit à plusieurs reprises qu'un plus grand nombre d'informations pourrait être repris dans les traces, notamment au niveau d'un chronométrage des actions que font les utilisateurs.
- Une idée qui nous paraît très intéressante concerne le choix des



paramètres qui établissent les dialogues mis en oeuvre par le programme à l'exécution. Nous avons vu que ces choix étaient faits avant l'exécution du programme de tenue des comptes, par l'éducateur à l'aide d'un programme auxiliaire. Ceci implique qu'en cours d'exécution du programme, il n'est plus possible de changer les paramètres, or il semble que ce soit une possibilité intéressante. On peut imaginer qu'il soit possible à tout moment de modifier les paramètres par le biais d'un menu déroulant. Cette action serait alors effectuée par l'éducateur, qui en cours de séance, interromperait l'utilisateur. A l'aide la souris, il choisirait dynamiquement le nouveau mode de dialogue.

Ceci est réellement réalisable et ne risque pas d'engendrer des erreurs. Il est, en effet, peu probable que la personne handicapée, change involontairement la valeur des paramètres au cours de ses manipulations. La souris de l'Amiga comporte deux boutons; en fait, celui de droite est réservé aux menus déroulants et n'est jamais utilisé dans la version actuelle du programme. Mais même s'il arrivait que l'utilisateur appuie sur le bouton de droite (ce que nous avons très rarement observé), il y a peu de chance, étant donné la séquence d'actions nécessaires, qu'il active le menu déroulant et sélectionne une nouvelle valeur pour un paramètre.

- Enfin, nous avons déjà vu qu'il peut être envisagé de fixer le déroulement du programme et non plus de laisser la liberté de choix à l'utilisateur (dès qu'il a atteint le menu). On pourrait avoir un paramètre qui permet à l'éducateur de fixer l'ordre du déroulement du programme, pour les personnes qui ont des difficultés à poser des choix.
- Une autre idée très motivante est venue lors de réunions avec les éducateurs qui s'occupent d'adultes handicapés mentaux présentant des problèmes auditifs. Il s'agit ici d'élargir le champ des dialogues. En effet, on peut imaginer que les questions posées à l'utilisateur le soient par le biais de dessins du langage des signes que ces personnes comprennent (alors qu'elles ne savent pas forcément lire). Bien entendu, il s'agirait encore d'une

possibilité à choisir grâce à un paramètre, c'est-à-dire que si l'éducateur souhaite voir apparaître les dessins du langage des signes, il choisit ce paramètre, et les phrases peuvent toujours apparaître ou non, selon son choix.

Dans le même ordre d'idée, les questions pourraient être posées par des dessins en BLISS, qui est un langage picto-graphique conçu pour les personnes ayant un handicap mental. Ces deux idées visent donc l'élargissement des modes de dialogues et par là-même du public atteint.

Ces quelques propositions pourraient être réalisées dans des travaux ultérieurs, en complément des autres parties du logiciel que nous avons présentées au chapitre 4, notamment la prévision budgétaire, l'épargne, la simulation d'achats, etc. Nous avons jugé utile d'en rendre compte ici afin que ces propositions, pour la plupart très intéressantes, ne soient pas oubliées.

#### 7.4. Perspectives au niveau de l'évaluation

##### 7.4.1. Plan d'expérimentation

Tout d'abord, on peut se demander si l'expérimentation telle que nous l'avons menée se rapproche d'un des designs proposés par Lynn Lyons Morris et Carol Taylor Fitz-Gibbon et présenté dans la section 6.1. Il faut indiquer que seule ce que nous avons appelé "l'évaluation pédagogique" se rapproche de l'évaluation préconisée par l'Université de Californie. Il s'agit, comme nous l'avons déjà dit, de mesurer les gains que peuvent réaliser les utilisateurs grâce au programme. Si l'on se réfère au tableau qui permet de choisir le design opportun (section 6.1.), on dira donc qu'on a uniquement un E-group et pas de C-group c'est-à-dire qu'on observe uniquement un groupe de personnes recevant le programme. La façon dont le questionnaire pédagogique est présenté (questions à répondre avant et après l'expérimentation) nous pousse à dire qu'un pretest et un posttest sont utilisés. Bien entendu, nous l'avons déjà signalé, il ne s'agit pas réellement de tests, mais d'appréciations faites par les éducateurs de ce que les handicapés comprennent. Tout cela nous permet de dire que

notre évaluation se rapproche du design 6, appelé à juste titre "the before and after design". Si l'on poursuit la lecture concernant ce design, on apprendra que c'est selon les auteurs le design le moins intéressant car il ne permet pas de voir les progrès qui auraient pu être faits sans l'utilisation du programme. Cela implique qu'on ne sait pas imputer au programme les progrès dont il est responsable. On perçoit à travers cette littérature, une volonté de quantifier à tout prix ces progrès. Ainsi, si l'on utilise le design 1 ou 3, on calculera le gain dû au programme par le calcul : Score du posttest du E-group - Score du posttest du C-group - Score du pretest du E-group (pour autant que le pretest du E-group et du C-group aient donné plus ou moins le même résultat). Si l'on utilise le design 6, on mesurera le gain (pas nécessairement dû au programme uniquement) par le calcul : Score du posttest - Score du pretest. Notre but à nous n'est pas quantifier absolument mais plutôt de rester au stade du qualitatif. L'éducateur, ayant vu les progrès réalisés par un handicapé peut dire si ceux-ci sont "supérieurs à" ou de nature différente de ceux qu'il aurait lui-même obtenus. Bien entendu, cela manque certainement de rigueur et c'est pourquoi nous nous limitons ici à tracer la voie pour des travaux ultérieurs concernant la mise sur pied d'une évaluation de logiciels en général et de programmes pour handicapés mentaux en particulier.

Nous allons maintenant essayer de donner quelques recommandations et spécifier les points qui devront être approfondis afin de mener une bonne expérimentation.

La première chose est de se définir une méthode d'évaluation, c'est-à-dire un design. On a vu que les chercheurs de l'Université de Californie donne le choix entre plusieurs designs en ce qui concerne l'évaluation d'un programme. Ces différents designs peuvent donc être d'application pour ce que nous avons appelé "l'évaluation pédagogique" de programmes pour handicapés mentaux. Il serait intéressant d'approfondir l'étude de ces neuf ouvrages (voir section 6.1.) afin de voir si une façon de faire adaptée au problème des handicapés peut être élaborée. Notons déjà que la randomization dans la population n'est pas très efficace car il y a trop de différence entre les individus. En effet, à moins de constituer des groupes d'énormément de personnes, il est clair qu'on ne parviendra pas à avoir deux groupes équivalents. Néanmoins, on pourrait donner un questionnaire à des éducateurs avant toute expérimentation afin qu'ils

répondent à des questions concernant les capacités et les caractéristiques des personnes handicapées dont ils s'occupent. On pourrait alors essayer de classer ces personnes en plusieurs groupes et par randomization, on formerait un E-group et un C-group dans chaque classe, c'est-à-dire que l'on mènerait une expérimentation par groupe. On répètera encore que les tests que l'on peut mener sont en fait des questionnaires remplis par les éducateurs ; il ne faut pas espérer de beaux tests comme ceux de mathématique où tout est précis et peut être facilement coté.

Il faut ensuite élaborer les grilles. On laissera le soin aux psychologues d'élaborer le questionnaire destiné à l'évaluation pédagogique. La construction de la grille pour l'évaluation de l'interface pourra être construite comme nous l'avons fait. Le tout est de disposer d'assez d'utilisateurs handicapés et de faire assez de séances avec eux afin de pouvoir se rendre compte de toutes les erreurs et incompréhensions qui sont possibles. Néanmoins, nous pensons que cette grille sera difficilement au point du premier coup. Ainsi, il est possible qu'une première itération donne une certaine grille dont les résultats après expérimentation montrent que certains problèmes sont survenus alors qu'on ne les avait pas prévus (ceci montre l'utilité de laisser sur le questionnaire des espaces libres afin que l'éducateur puisse noter ce genre de choses). Grâce à cela, il sera alors possible de mettre sur pied une grille définitive qui pourra servir pour une nouvelle expérimentation (qui sera peut-être à plus grande échelle). Nous avons déjà dit que souvent, il est n'est pas intéressant de n'avoir pour réponses possibles que OUI et NON, mais bien une échelle comportant davantage de possibilités (0 à 10 par exemple). En effet, plusieurs fois dans notre évaluation, l'éducateur indiquait pour certaines questions des précisions du genre "un peu", "plus ou moins", "pas tellement". Cela montre bien qu'une échelle aurait été nécessaire.

Ensuite, on regardera quelles sont les traces des manipulations qu'il est utile et possible de garder. Utile d'abord puisqu'il n'est pas toujours nécessaire de garder le maximum de choses et possible ensuite puisqu'il ne faut pas oublier que l'on reste toujours tributaire des possibilités de la machine. On essaiera de faire faire à la machine le maximum de choses, ce que nous n'avons pas fait nous mêmes. Ainsi par exemple, on pourra déterminer des types d'erreurs et que la trace livre directement le nombre d'occurrences de chaque type. On pourra surtout faire en sorte que

l'ordinateur calcule lui-même les transitions, telles que nous les avons présentées dans le point 6.3.1.

Quand le design a été choisi, les questionnaires élaborés et la trace préparées, on peut passer à l'expérimentation proprement dite. Un problème est de savoir combien de temps elle doit durer. Nous ne pouvons y répondre, mais nous pensons que six mois ne seraient pas excessifs. Il est clair cependant que cela dépend de la complexité du programme en question. Un autre point que nous avons déjà soulevé est la subjectivité toujours possible des éducateurs. Dès lors, on essaiera d'avoir un nombre assez élevé d'éducateurs afin que celle-ci soit amoindrie. Si ce n'est pas possible, on demandera à l'éducateur d'indiquer son nom sur chaque questionnaire pour pouvoir déceler les questions auxquelles il répond toujours de la même façon, etc.

Quand on a reçu les questionnaires remplis et les traces des manipulations, on pourra exploiter les résultats. La première chose est d'interpréter les traces et, si elles ont été bien faites (transitions calculées automatiquement), ce ne sera pas un travail fastidieux. En ce qui concerne le questionnaire destiné à l'évaluation de l'interface, on pourra calculer les fréquences des résultats comme on l'a expliqué plus haut. Si l'on a mis des échelles de 0 à 10 par exemple, on pourra faire la moyenne des scores sans oublier d'observer toujours les scores effectifs (il est en effet possible qu'un score de 5 provient de 1000 scores à 0 et de 1000 scores à 10, ce qui voudrait certainement dire beaucoup de choses). L'exploitation de la grille destinée à l'évaluation pédagogique sera faite par les psychologues qui détermineront quelles sont les lacunes du logiciel et, avec les informaticiens, y trouver des solutions.

#### 7.4.2. L'exploitation des traces et des questionnaires

Nous avons vu que l'objectif de ce travail était de présenter la problématique du développement de projets informatiques destinés à des utilisateurs avec un handicap mental, en nous basant sur le projet concret auquel nous avons collaboré et en exposant la démarche que nous avons suivie lors des différentes phases. Il ne s'agissait donc pas d'une méthodologie à suivre pour travailler dans le domaine de l'informatique et du handicap mental. Nous avons vu aux sections 7.1. et 7.2. d'évidentes

faiblesses de notre démarche. Dès lors on peut envisager un travail de réflexion plus théorique, visant à proposer une méthodologie pour la phase d'évaluation de projets destinés à des personnes handicapées. C'est un domaine où très peu de choses sont faites (même pour des applications classiques). Nous avons réellement été confrontés à un vide en la matière et il y a là beaucoup d'espace pour la recherche : que ce soit au niveau des outils, des moyens utilisables, ou au niveau de la méthode pratique de mise en oeuvre d'expérimentations et d'exploitation des traces de manipulations et des questionnaires, qu'il s'agisse de l'évaluation des fonctionnalités et des interfaces ou de l'évaluation pédagogique de l'utilisation de programme pour handicapés mentaux.

Le domaine de l'informatique et du handicap mental est encore trop largement inexploré. Il offre beaucoup de voies de recherche, et la demande en logiciels est réelle.

Pour que des projets aient la possibilité d'être acceptés par les utilisateurs, ils doivent être menés par des équipes pluridisciplinaires comprenant des informaticiens, des psychologues et des éducateurs qui connaissent les problèmes concrets de la vie en institution, et finalement les utilisateurs potentiels.

Mais, en plus de projets répondant à des demandes ponctuelles, il y a place pour des travaux plus théoriques visant à proposer des méthodologies, notamment au niveau de l'évaluation de logiciel.

De toute évidence, c'est un domaine qui mérite qu'on y consacre beaucoup d'efforts et nous pouvons dire qu'il apporte aussi beaucoup de satisfactions.

## **Bibliographie**

- Card, S.K., Moran, T.P. et Newell, A. (1980). **The keystroke-Level Model for User Performance Time with Interactive Systems**, Communications of the ACM 23 (7)
- Card, S.K., Moran, T.P. et Newell, A. (1980). **Computers Text-Editing : an Information Processing Analysis of a Routine Cognitive Skill**, Cognitive Psychology 12
- Card, S.K., Moran, T.P. et Newell, A. (1983). **The psychology of Human-Computer Interaction**, Hillsdale, N.J.: Lawrence Erlbaum Associates
- Chandelon, Murielle et Warnant, Geneviève. (1987). **Modélisation d'une application interactive**, Mémoire présenté en vue du titre de Licencié et Maître en Informatique, FNDP, Septembre 1987
- Chernicoff, Stephen. (1985). **Macintosh TM Revealed**, Volume one: Unlocking the toolbox, Hayden, Appel Press, Hasbrouck Heights, New Jersey, USA, 1985
- Chevaux, D. et Lepretre, P. **Emploi de l'ordinateur pour les enfants et adolescents en difficulté (approche psychopédagogique)**, Dossiers de l'association pédagogie médicale appliquée
- Courbin, Pierre. (1982). **L'informatique : aide ? ou handicap ?**, Informatique et handicaps, Les cahiers de la fondation R. Bull, 4 (décembre, 1982)
- Coutaz, Joelle. (1986a). **La construction d'interfaces homme-machine**, Rapport de recherche, Novembre 1986
- Coutaz, Joelle. (1986b). **The construction of User Interfaces and the object paradigm**, ECOOP1987

- Draper, (1984). **Software Engineering for User Interfaces**, Processing of the 7th International Conference on Software Engineering, Orlando, FL., March 1984
- Edmonds, E.A. (1981). **Adaptive Man-Computer Interfaces**, in Computing Skills and the User Interface, Coombs, M.J. & Alty, J.L. (eds) Academic Press, London
- Gabus, Jean-Claude. **Aides techniques, voix synthétiques et communication des personnes sans langage verbal, 'le point de la situation'**, Fondation Suisse pour les Téléthèses, février 1989
- Gleaves, Richard (1984). **Modula-2 for Pascal Programmers** Springer-Verlag
- **Intuition Reference Manual**, Commodore Business Machines, Inc
- Konsynski, Kuo & Kuo, Bob. (1985). **An architecture for Dialogue Management : Implications in User-Computer Dialogue Design**, University of Arizona; Interfaces in Computing, 3
- Mortimore, Eugene P. (1987). **Amiga Programmer's Handbook**, Volume I et II, Second edition, SYBEX
- Noirhomme-Fraiture, Monique (1989). **L'analyse en composantes principales et l'analyse des correspondances multiples. Application à l'étude des grilles d'évaluation**, RP10/89, HANTEPSYCOM, Bxl 6-7 juillet
- Norman, D.A. (1984). **Stages and Levels in Human-Machine Interaction**, International Journal of Man-Machine Studies, 21
- Petoud, Isabelle. (1987). **Conception de l'interface homme-machine**, Lausanne, session de printemps 87 E.H.E.C. Université de Lausanne



- Rosenthal, Dave & Yen, Albert. (1983). **User Interface Models Summary in graphical Input Interaction Technique : Workshop summary**, Computer Graphics; January 1983
- Ruche, Jean-Claude. (1988). **Méthodologie de spécification d'une interface homme-machine**, Mémoire présenté en vue du titre de Licencié et Maître en Informatique, FNDP, Septembre 1988
- Schneiderman, Ben. (1987). **Designing the User Interface : Strategies for Effective Human Computer Interaction**, Addison-Wesley Publ. Comp., 1987
- Sproull, (1983). **Programming the User Interface in Man-Machine Interaction**, Proceedings of the joint Conference IBM/University of Newcastle upon Tyne Computing Laboratory, September 1983
- Sufrin, Bernard. (1986). **Formal methods and Design of Effective User Interface**, in People and Computers : Designing for Usability, Proceedings of the Second Conference of the British Computer Society Human Computer Interaction Specialist Group, British Computer Society Workshop Series
- Taylor Fitz-Gibbon, Carol & Lyons Morris, Lynn (1988). **How to Design a Program Evaluation**, Center for the Study of Evaluation, University of California Los Angeles, Sage Publications, second printing, 1988
- U.N.A.P.E.I. de Montbeliard (1987). **Special colloque Lons : Libres propos**, Lettre "Informatique et éducation", 6 (décembre, 1987)